



Version 1.0

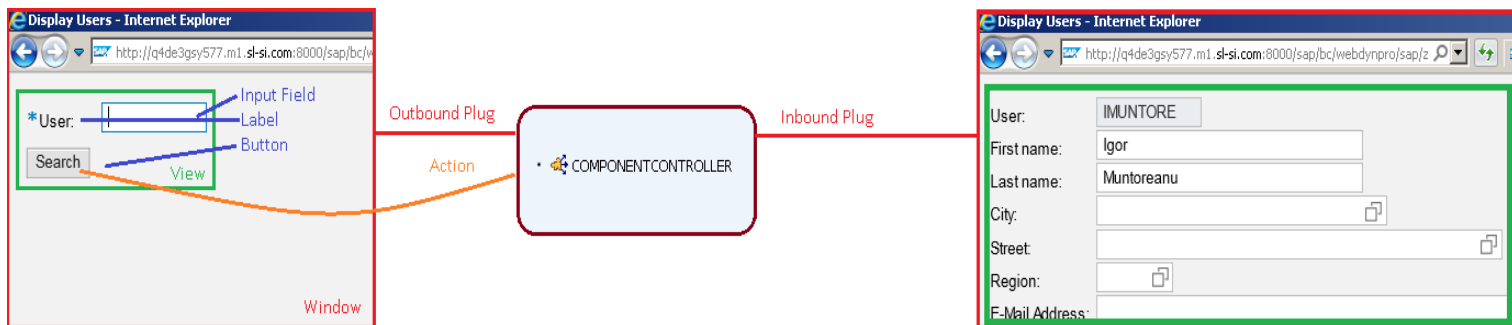
Date 27.06.2016

Status In Development

Number 1

Initial version by Igor Muntoreanu

## Exercise Review:



## Practical Guide:

- 1) Go to SE80 (that is the transaction that you are going to manipulate all the WebDynpro ABAP components) and create a WebDynpro Component:

When you create a WD Component, a Window and a View are required to be generated too.

Web Dynpro: Component / Create Interface

Name: ZWDC\_USER\_DISPLAY

Description: Display User Contact Information

Type: ☒ Web Dynpro Component ☐ Web Dynpro Component Interface

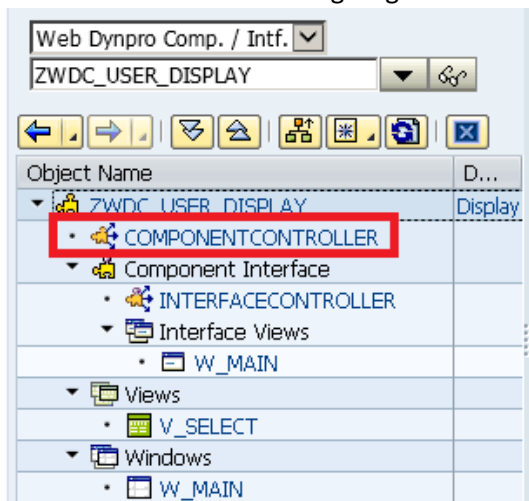
Window Name: W\_MAIN

View Name: V\_SELECT

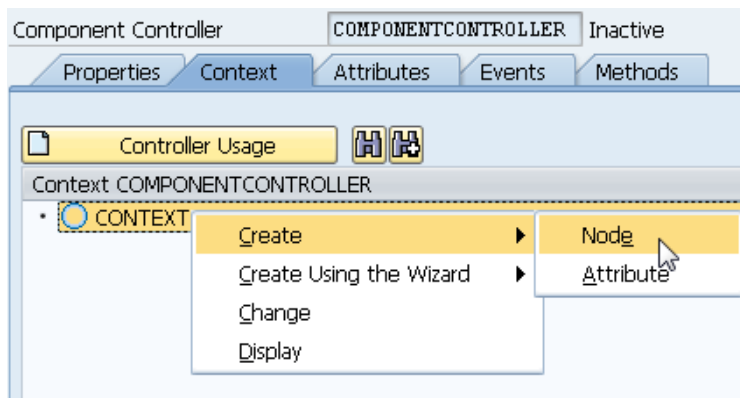
- 2) **Manipulating the Component Controller**

The Component Controller is always just one. It drives the functionality of the entire component. When starting a WebDynpro application, the component controller is instantiated by the WebDynpro runtime.

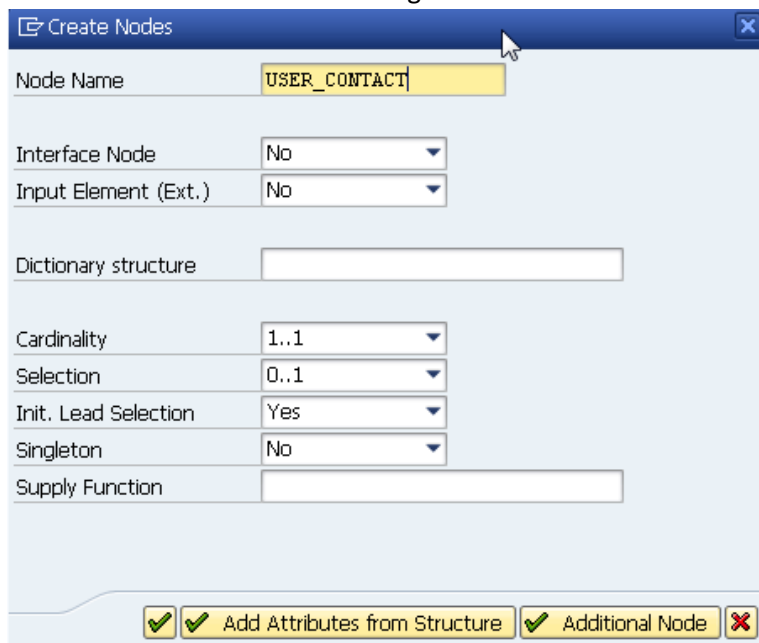
- a. Double click on the following target:



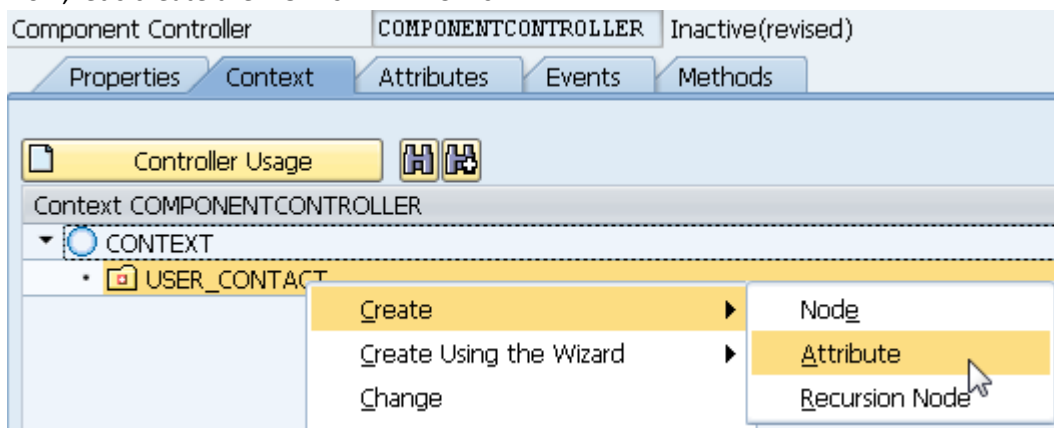
- b. Inside the context of the COMPONENTCONTROLLER, tab Context, create a NODE:  
NODE is a group of Attributes all together.  
CONTEXT is the place where you defined what NODES/ATTRIBUTES the current Component is going to use.



- c. Create the NODE like the following Screen Shot:



- d. Now, let's create the NODEs ATTRIBUTES:

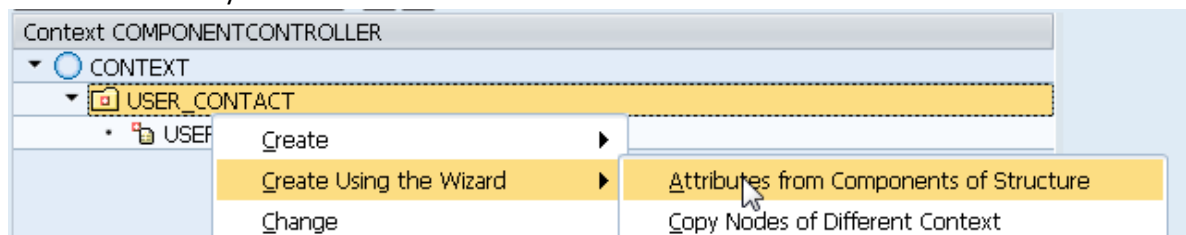


**Create Attribute**

Attribute Name	USERNAME
Type assignment	Type
Type	XUBNAME
Read-only	No
Default Value	
Input Help Mode	Automatic
Determined Input Help	
Type of Input Help	

☒ ☒ Additional Attribute ☐

You can create one attribute at time. In this case, let's use the WIZARD to bring all the attributes we want automatically:

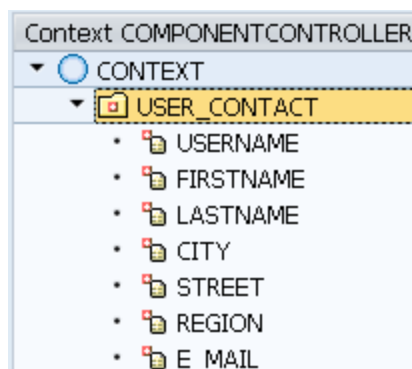


**Add Attributes to Node USER\_CONTACT**

Structure Name	BAPIADDR3
----------------	-----------

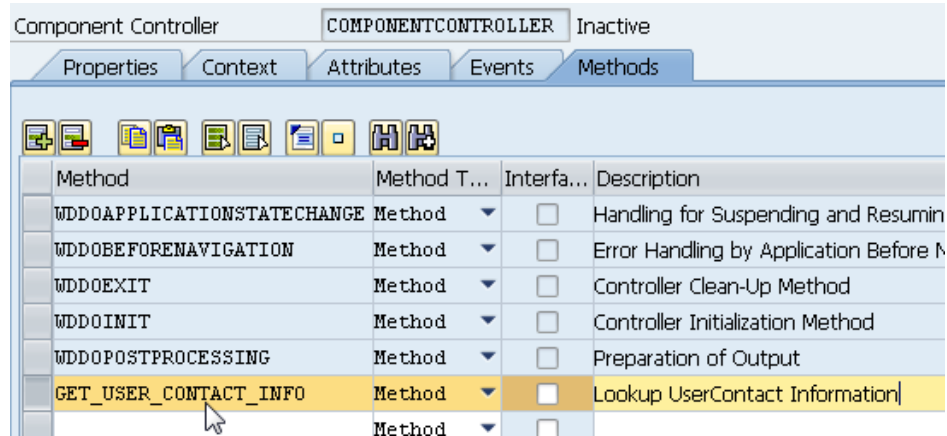
☒ ☐

Select the fields and bring the following ATTRIBUTES:



- e. Create a Lookup Method in the “Methods” tab and implement the following code ( Do not pay attention to the code now, just copy and paste for while):

The methods that you can create here can be called anywhere in the next Views.



```

DATA: lo_nd_user_contact TYPE REF TO if_wd_context_node,
      lo_el_user_contact TYPE REF TO if_wd_context_element.

DATA: lv_username TYPE REF TO wd_this->element_user_contact-username.

DATA: ls_address TYPE bapiaddr3,
      lt_return  TYPE TABLE OF bapiret2.

DATA: ls_user_contact TYPE wd_this->element_user_contact.

" Read the selected user name into context:
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( ).

lv_username ?= lo_el_user_contact->get_attribute_ref('USERNAME').

" Lookup the user's contact information:
CALL FUNCTION 'BAPI_USER_GET_DETAIL'
  EXPORTING
    username = lv_username->*
  IMPORTING
    address  = ls_address
  TABLES
    return   = lt_return.

" Copy the results to a structure and bind it to the context:

ls_user_contact-username      = lv_username->*.
ls_user_contact-firstname    = ls_address-firstname.
ls_user_contact-lastname     = ls_address-lastname.
ls_user_contact-street       = ls_address-street.
ls_user_contact-city         = ls_address-city.
ls_user_contact-region       = ls_address-region.
ls_user_contact-e_mail       = ls_address-e_mail.

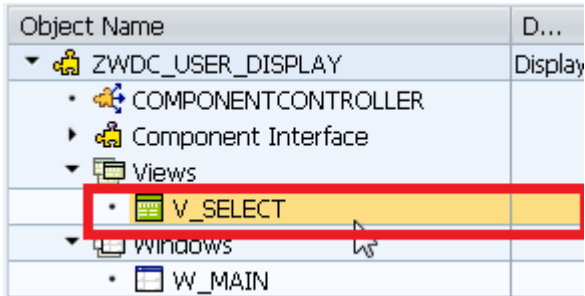
" Bind it to the context
lo_el_user_contact->set_static_attributes( ls_user_contact ).

```

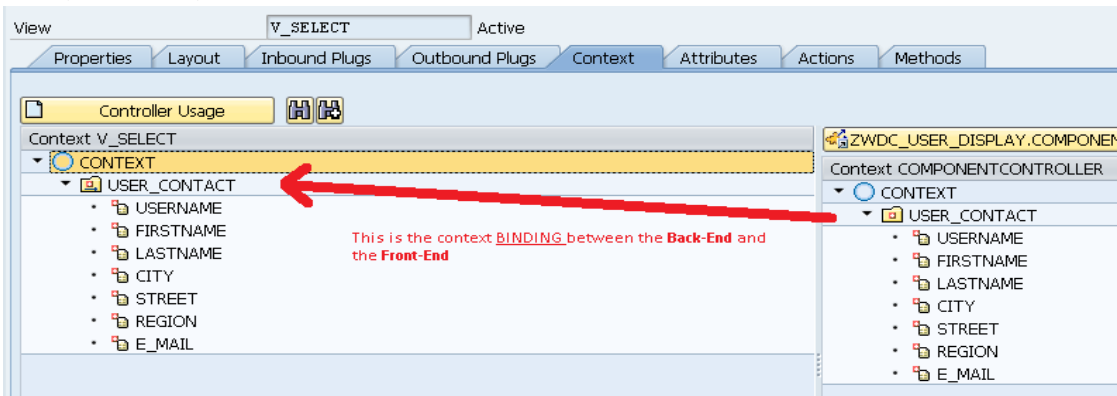
### 3) Laying out the Selection View – Front End:

The View is the place where you create the Front End Webpage.

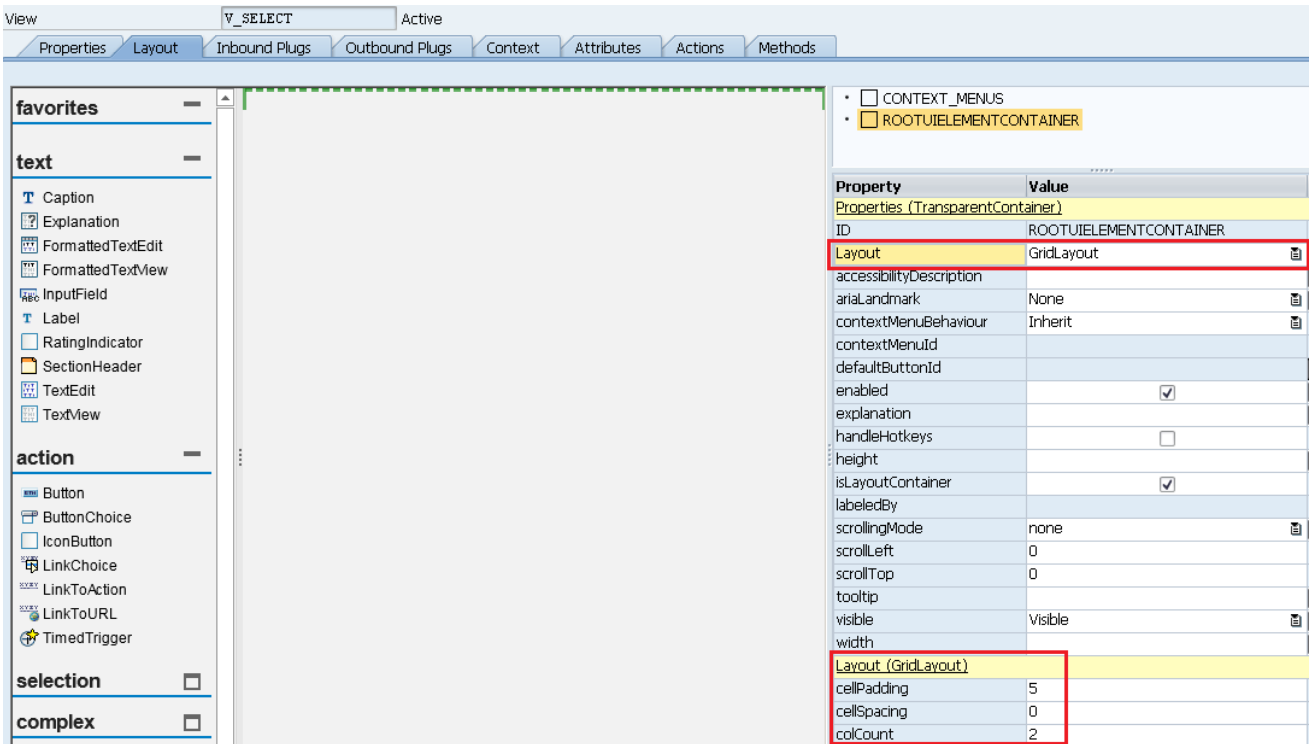
- a. Double Click on the V\_SELECT view and go to the “Context” tab:



- b. Drag and drop the NODE from the COMPONENTCONTROLLER(on the right) to the Context of the View(on the left).



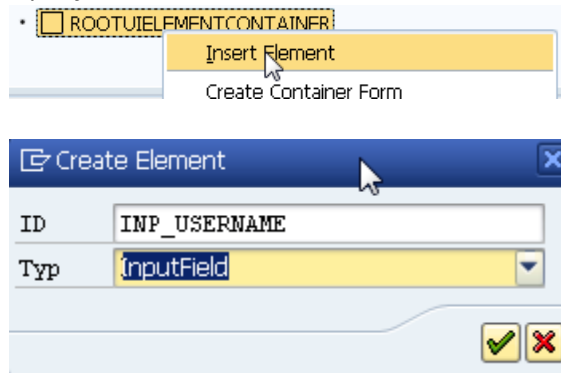
- c. Go to the “Layout” tab, set the ROOTUIELEMENTCONTAINER as displayed below:



In the “Layout” tab you will be able to define all the Webpage Screen Elements and their layouts.

So, let's create an Input field, a Label and a Button:

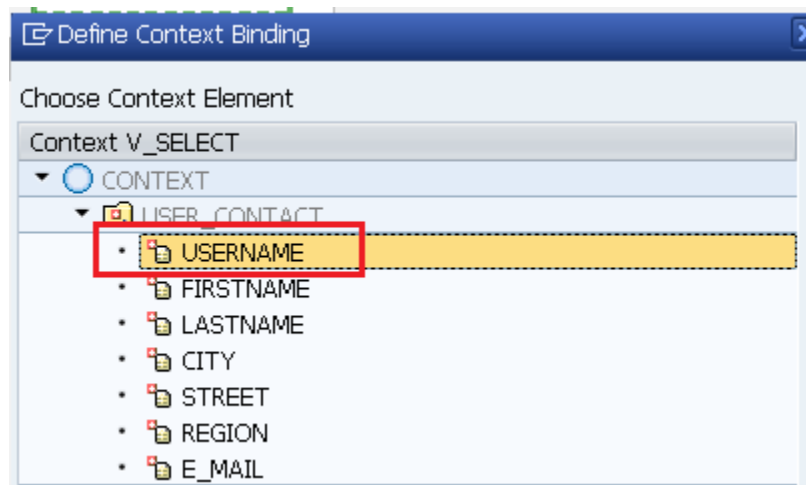
- *Input field:*



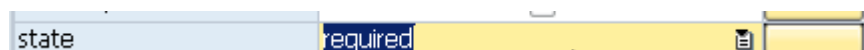
Bind the “value” Property:

Property	Value	Binding
suppressValueHelp	<input type="checkbox"/>	
textDirection	Inherit	
tooltip		
value		
visible		

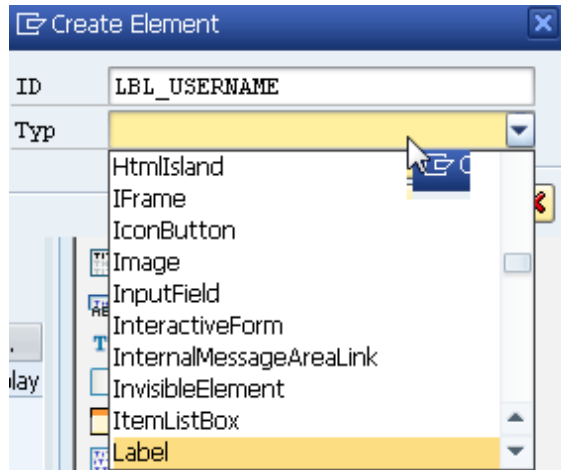
Choose the USERNAME attribute from the Context Element



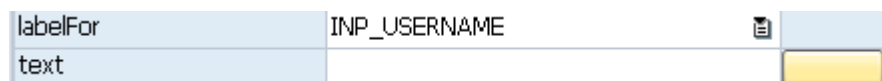
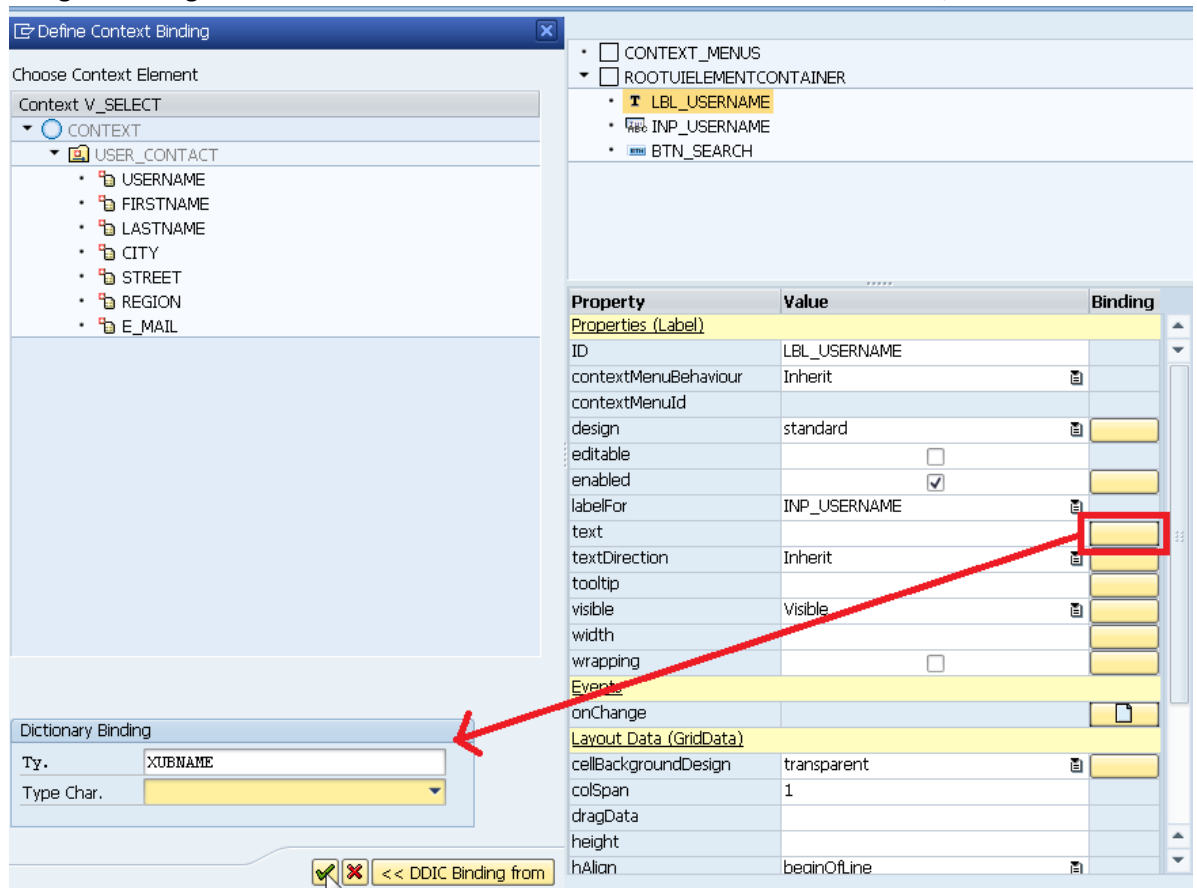
Make the field required:



- *Label for the Input field:*

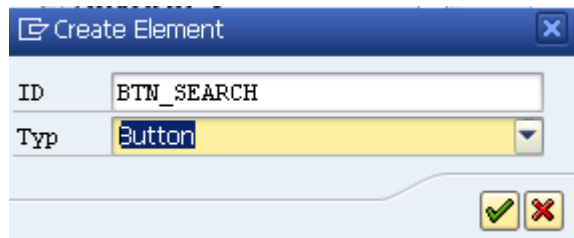


Links the Label with the Input field in the property “labelFor”. Give a text to the label linking it by using the Binding with a Data Element. It will take the text direct from the Table/Structure:





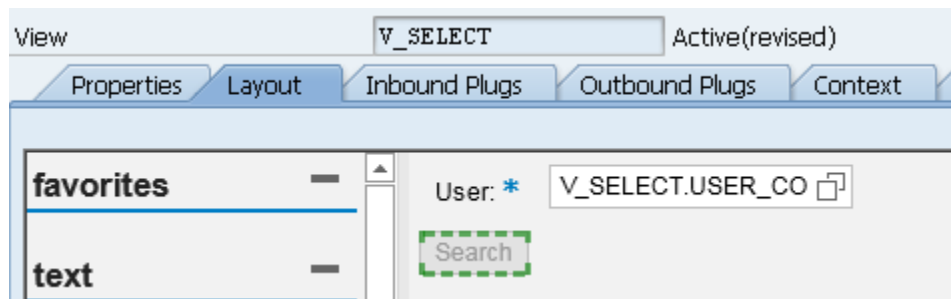
- **Button:**



Give a text to the Button



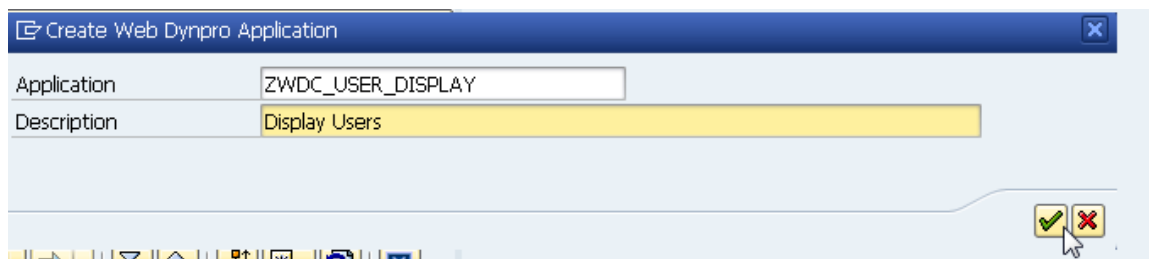
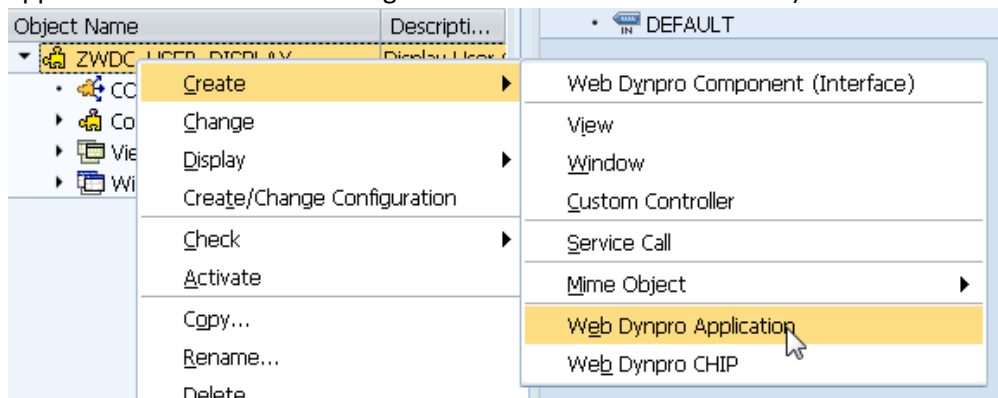
The final result of the view layout should be like this:



#### 4) Create the WebDynpro Application

Analogously, it is like generating a transaction for a Report.

- Create the WebDynpro Application like showed below (it generates automatically an URL for the application if all the BASIS configuration in SCIF transaction are ok).



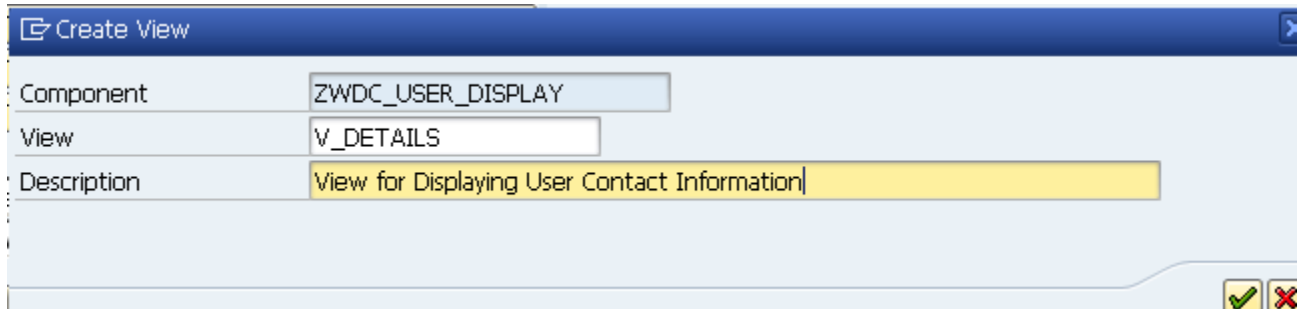
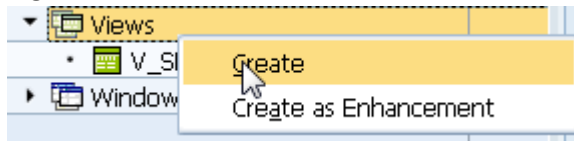
Make sure to Save and Activate your development times in times:



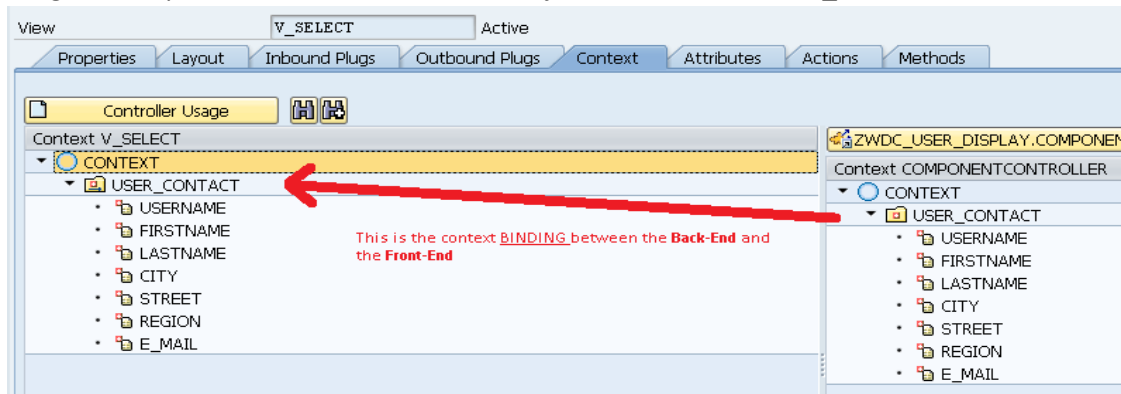
As you can see that is no functionality yet. The next steps will create them.

## 5) Laying out the Details View – Front End:

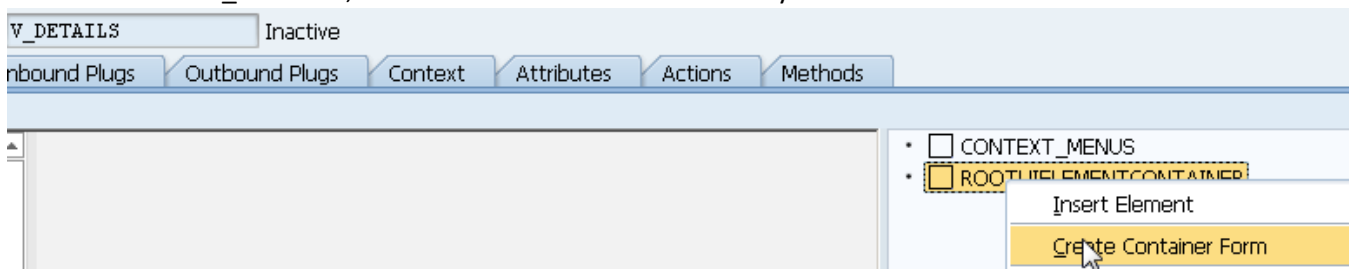
- a. Right click on “Views” and create a new one like showed below:

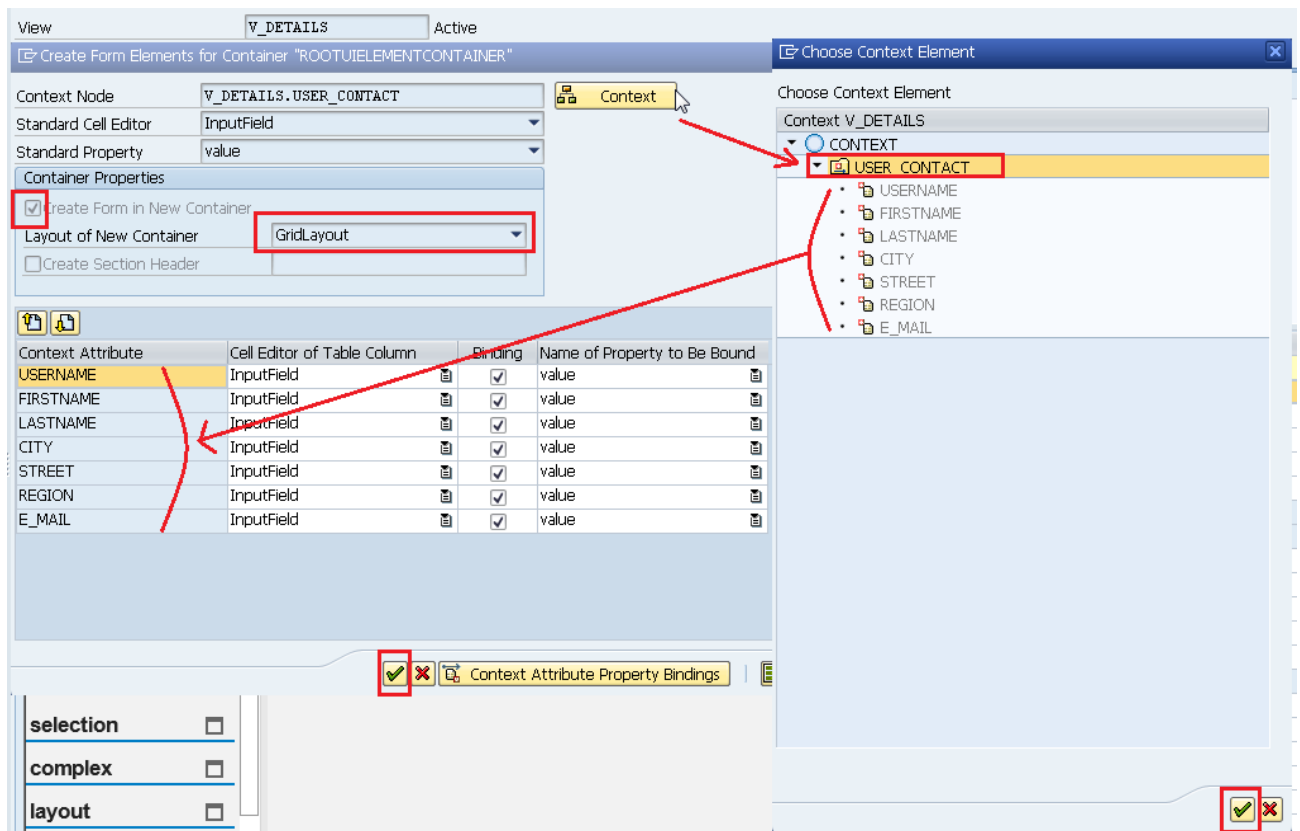


- b. Drag and Drop the context for this view also, just like we did in the V\_SELECT view:

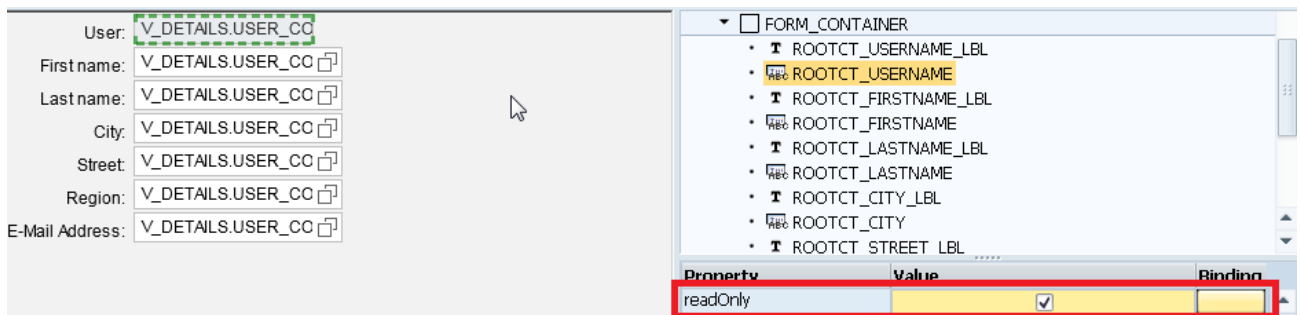


- c. Inside the view “V\_DETAILS”, create a “Container Form” in the layout tab:

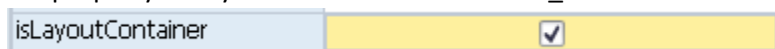




Set the Input Field "USERNAME" as read-only:

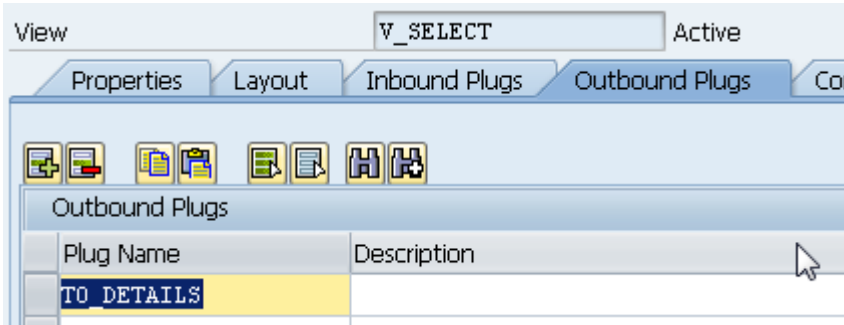


Set property "isLayoutContainer" from FORM\_CONTAINER marked:

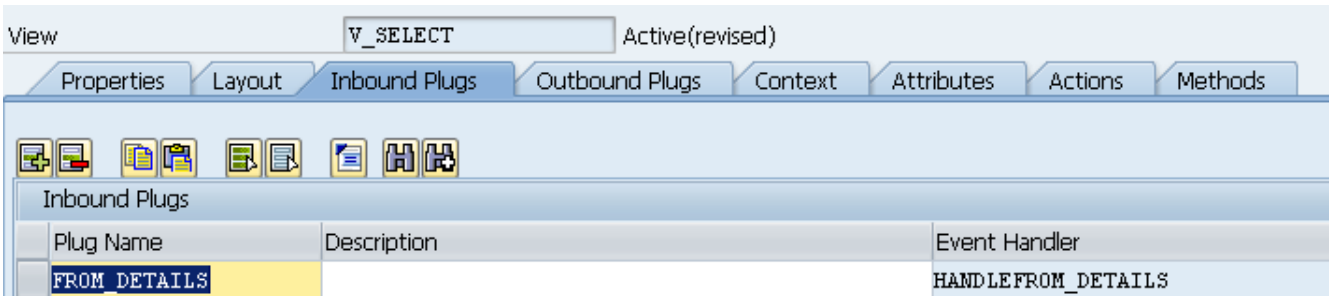


**6) Outbound Plug in the V\_SELECT View (Plugs are nothing more than Navigation specification between the Views).**

- a. Open the V\_SELECT View and in the tab Outbound Plugs and create a Plug like below:

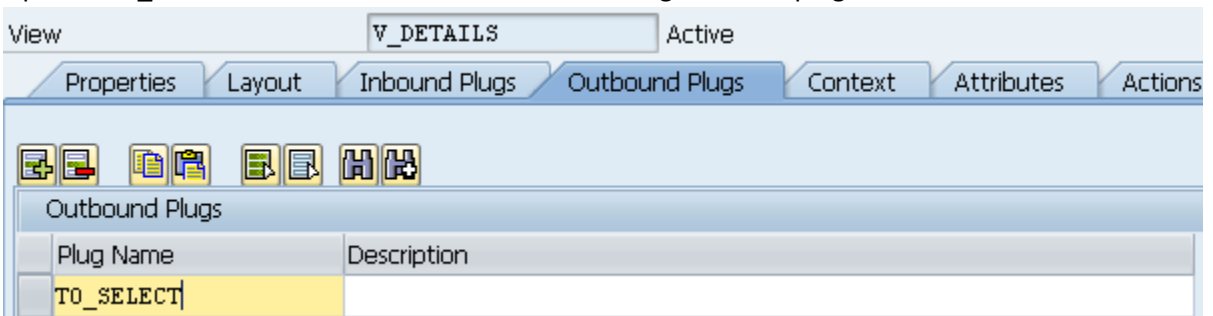


- b. Now, open the Inbound Plugs and create like below:

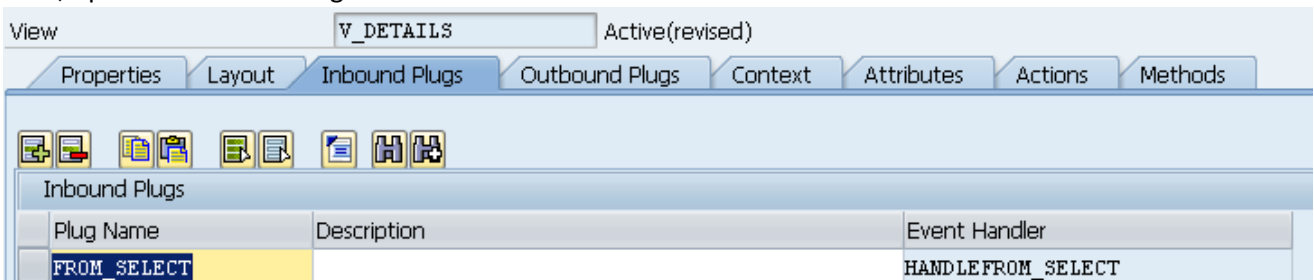


**7) Inbound Plug in the V\_DETAILS View**

- a. Open the V\_DETAILS View and in the tab Outbound Plugs create a plug like below:

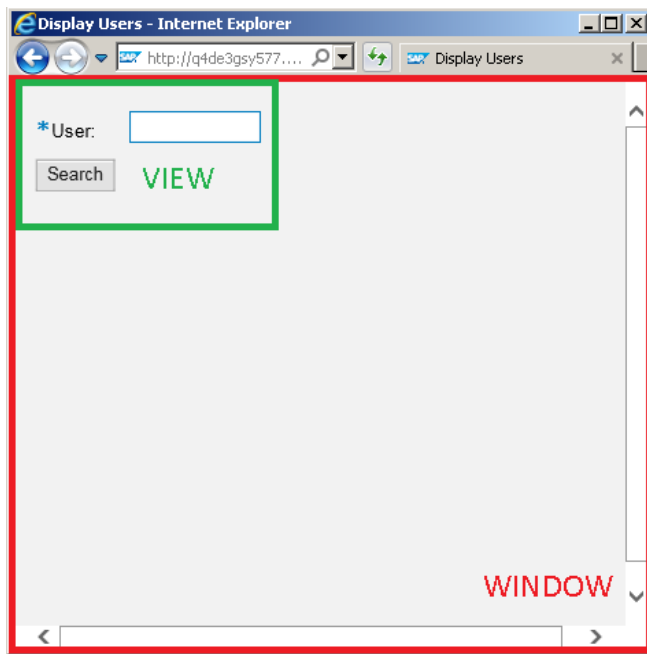


- b. Now, open the Inbound Plugs and create like below:



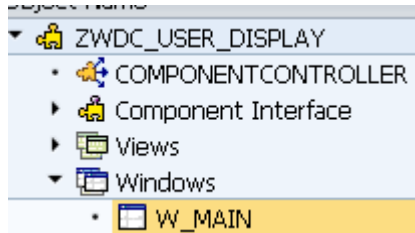
**8) Window Manipulation**

A Windows is different from the view, take a look on the following SS:

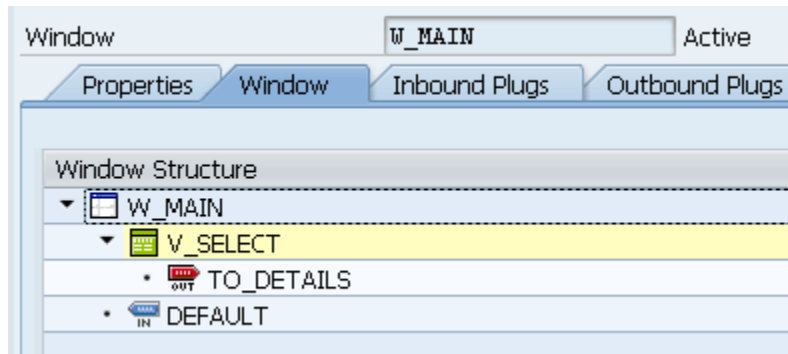


The Window can have a lot of Views together.

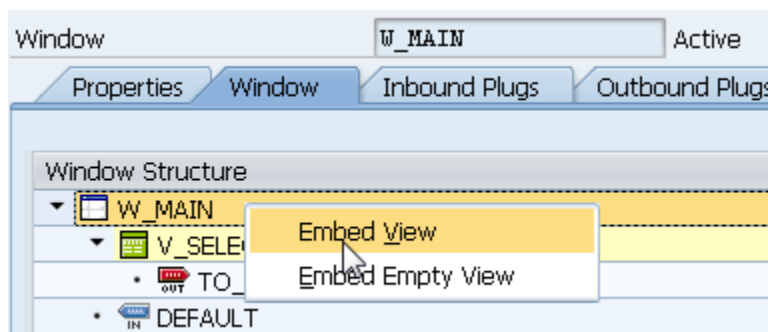
- a. Go to the Main Window:



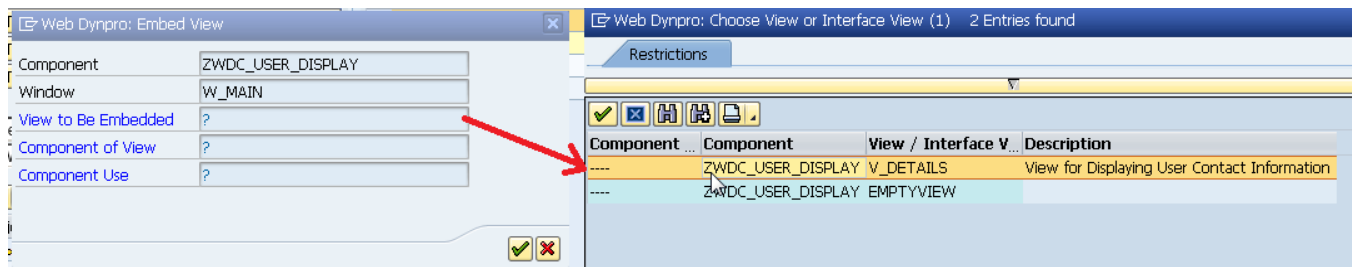
- b. Here we need to create the relations between the Views via the Inbound/Outbound Plugs:



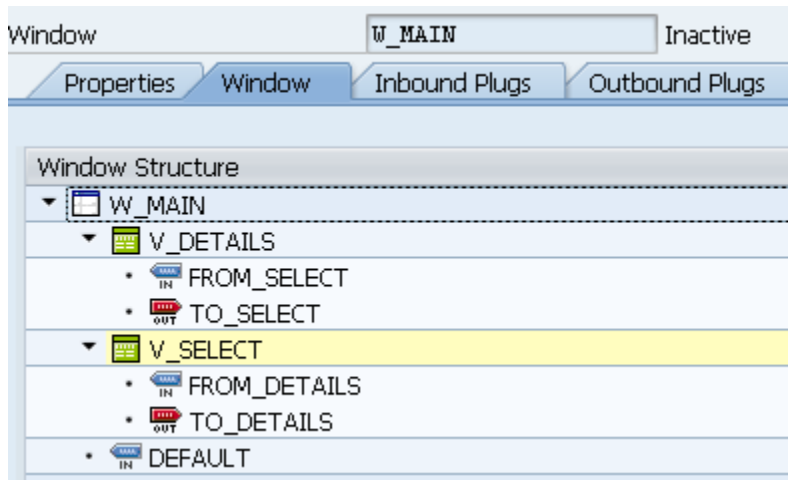
Right click on W\_MAIN and choose "Embed View":



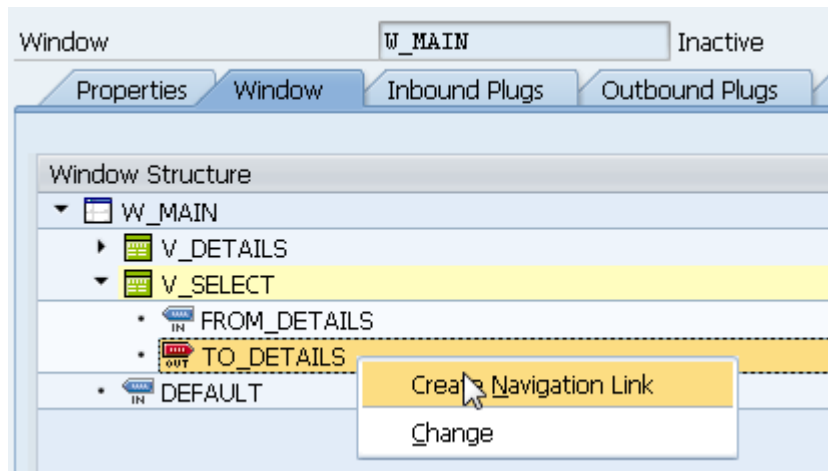
Select the V\_DISPLAY view to be Embedded like the SS below:



Your Window configuration will be like this:

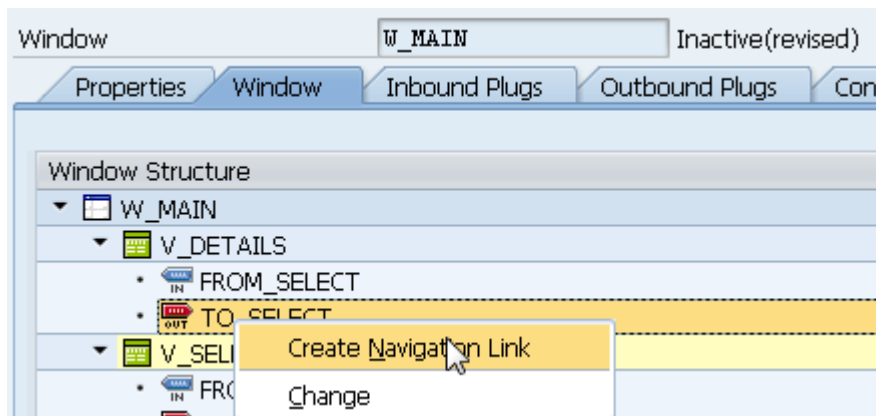


Now its necessary to create the link between the Plugs, by creating a Navigation Link:



Web Dynpro: Choose Destination for Navigation

Start Component	ZWDC_USER_DISPLAY
Start View	V_SELECT
Outbound Plug	TO_DETAILS
Dest.View	V_DETAILS
Embedding Position	W_MAIN
Inbound Plug	FROM_SELECT



Web Dynpro: Choose Destination for Navigation

Start Component	ZWDC_USER_DISPLAY
Start View	V_DETAILS
Outbound Plug	TO_SELECT
Dest.View	V_SELECT
Embedding Position	W_MAIN
Inbound Plug	FROM_DETAILS

## 9) Create an ACTION to the SEARCH Button.

- Enter in the V\_SELECT View and Click 2x In the SEARCH Button:

View V\_SELECT Active

Properties Layout Inbound Plugs Outbound Plugs Context Attributes Actions Methods

favorites

text

Caption

User: \* V\_SELECT.USER\_CO

Search

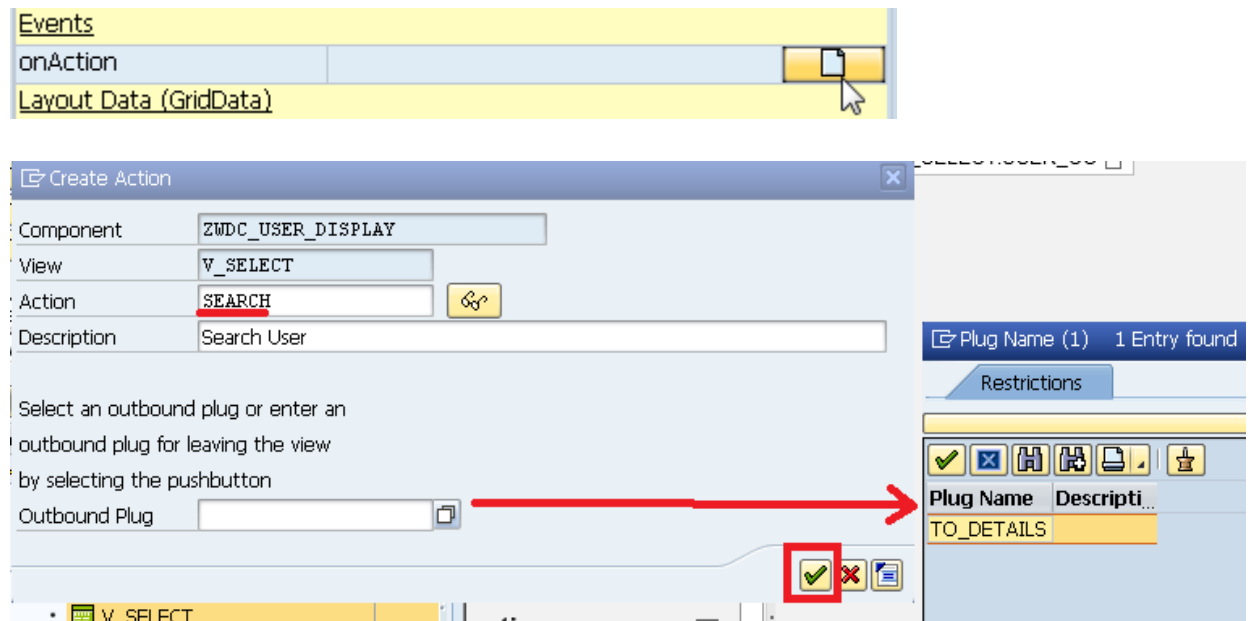
CONTEXT\_MENUS

ROOTUIELEMENTCONTAINER

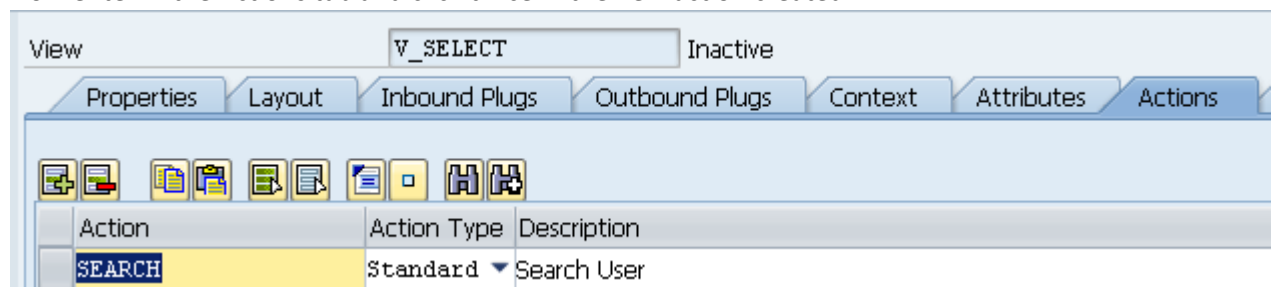
- LBL\_USERNAME
- INP\_USERNAME
- BTN\_SEARCH



Create a new Action for the button:



Now enter in the Actions tab and click twice in the new action created:

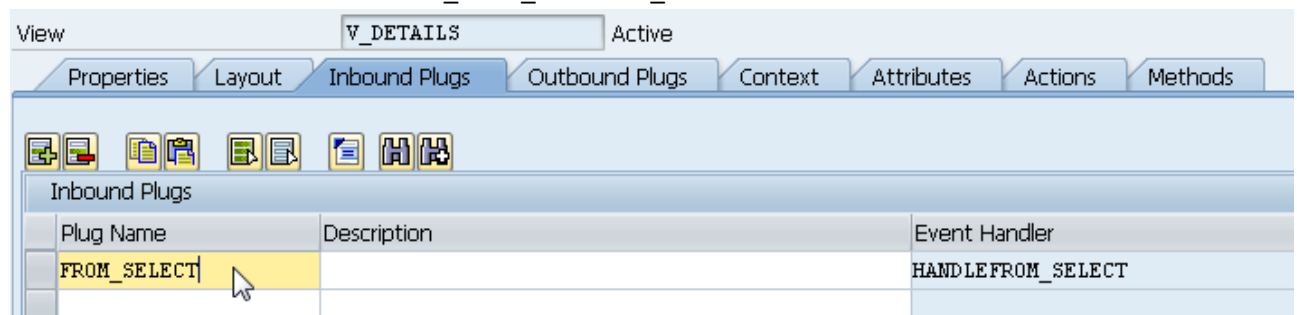


The coding brings automatically:

```
method ONACTIONSEARCH .
    wd_this->fire_to_details_plg(
    ).
endmethod.
```

## 10) Coding in the View V\_DETAILS under Inbound Plug

- We will call the created method GET\_USER\_CONTACT\_INFO



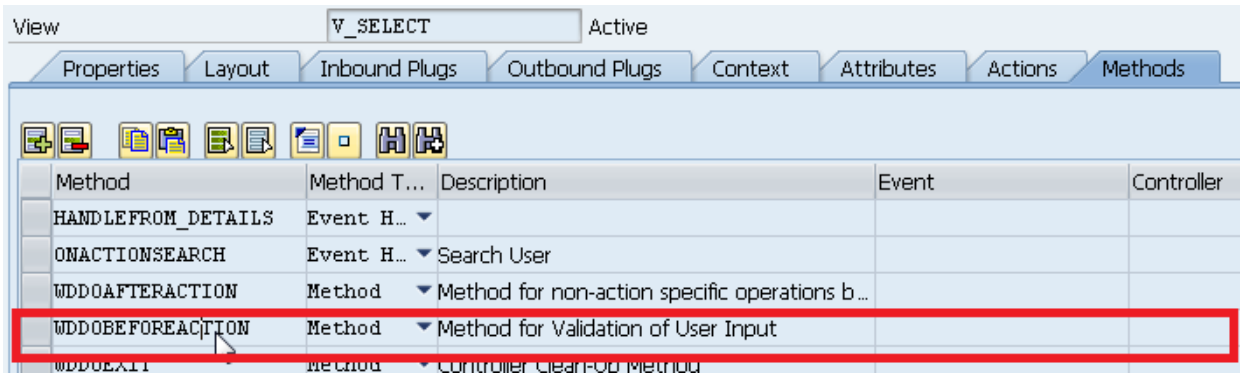
```

1  METHOD handlefrom_select .
2
3      " Invoke the component controller method to load the user contact info:
4
5      wd_comp_controller->get_user_contact_info( ).
6
7  ENDMETHOD.

```

### 11) Using a Pre-Defined Method to promotes validations.

- In our case despite of the field USERNAME of the view V\_SELECT is mandatory, it is necessary to create a validation for that differently from common Reports for example.
- Open the V\_SELECT view and select WDDOBEFOREACTION:



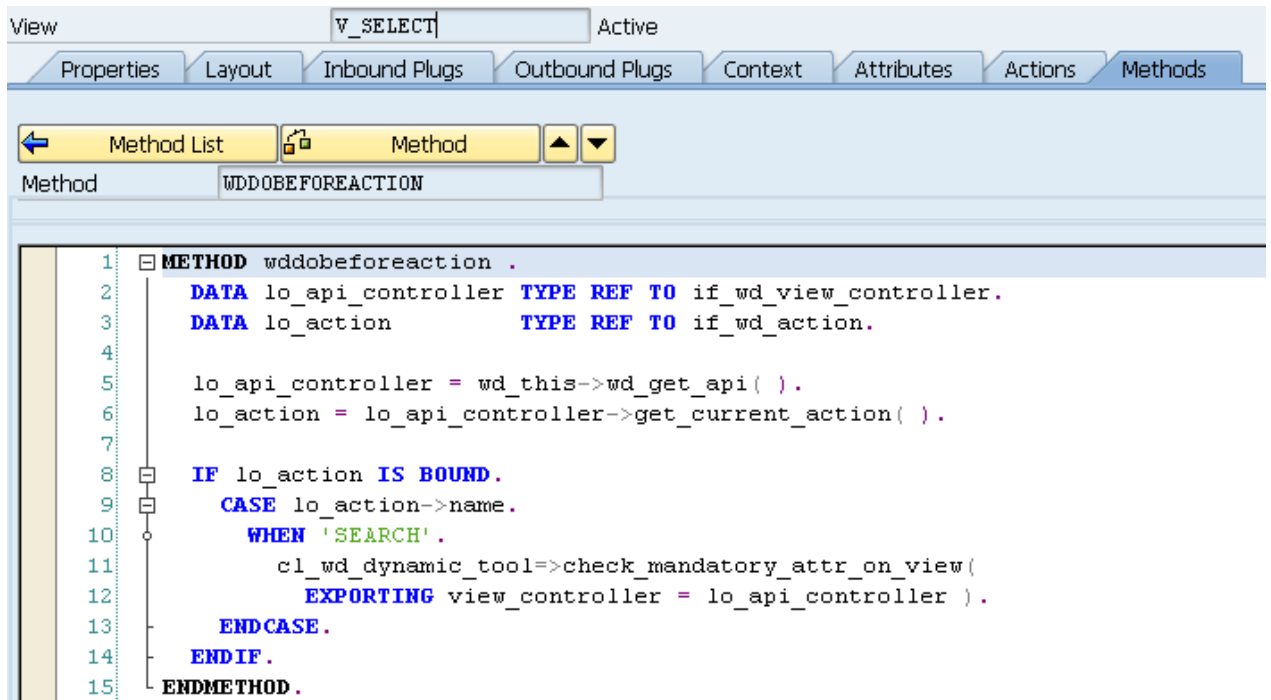
- As you can see a pre-code is already defined. We will take it:

```

1  method WDDOBEFOREACTION .
2
3      * data lo_api_controller type ref to if_wd_view_controller.
4      * data lo_action         type ref to if_wd_action.
5
6      * lo_api_controller = wd_this->wd_get_api( ).
7      * lo_action = lo_api_controller->get_current_action( ).
8
9      * if lo_action is bound.
10     *   case lo_action->name.
11     *       when '...'.
12
13     *   endcase.
14     * endif.
15 endmethod.

```

- d. The new code will be like that:

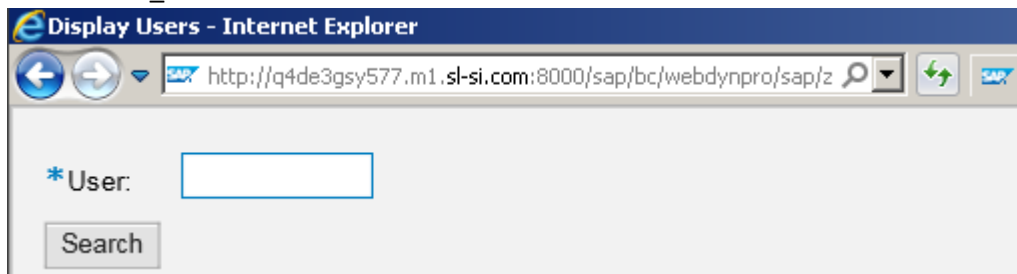


- 12) The WebDynpro is finally done, let's check it out:

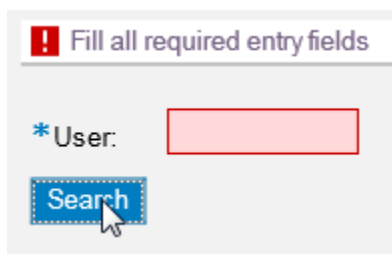
- a. Go to the WDA and run it.



- b. This is the V\_SELECT view



Click on Search without typing the user:



Now put some valid value and execute:

\*User:

This is the V\_DETAILS view.

Display Users - Internet Explorer

http://q4de3gsy577.m1.sl-si.com:8000/sap/bc/webdynpro/sap/z

User:

First name:

Last name:

City:

Street:

Region:

E-Mail Address:

### 13) Debugging Tool Tips:

- Put an external break-point in the method WDDOBEFOREACTION

Component Controller **COMPONENTCONTROLLER** Active

Properties Context Attributes Events **Methods**

Method List

Method

Exceptions

Parameter	Decl...	RefTo	Opt	Associated Type	Default Value	Description
		<input type="checkbox"/>	<input type="checkbox"/>			
		<input type="checkbox"/>	<input type="checkbox"/>			
		<input type="checkbox"/>	<input type="checkbox"/>			

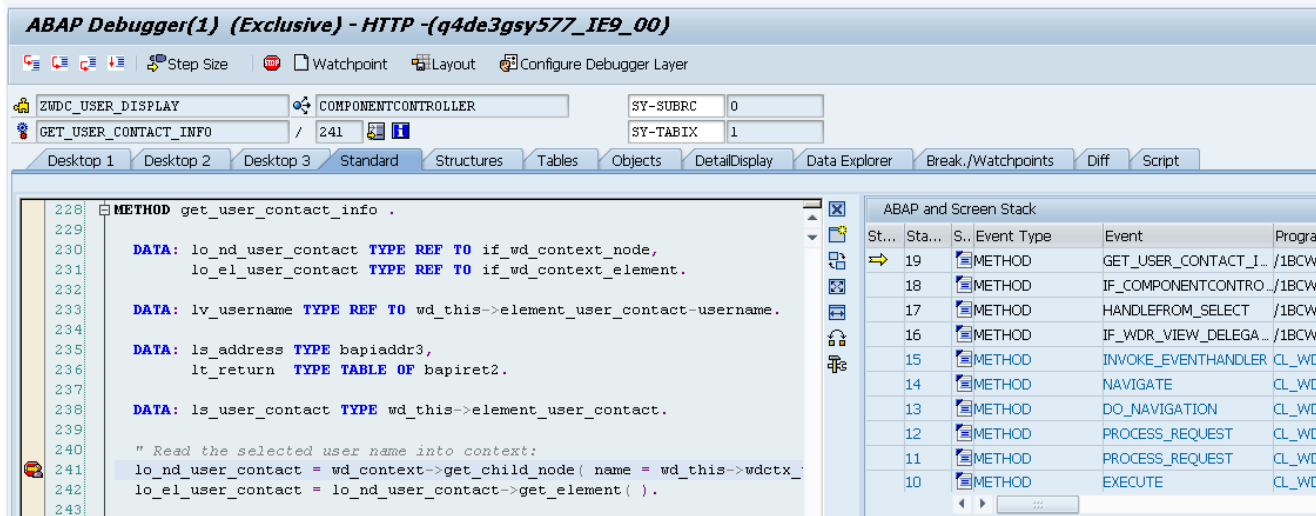
14 | 15 |

```
lo_nd_user_contact = wd_context->get_child_node( name = wd_this->wdctx_user_contact ).
```

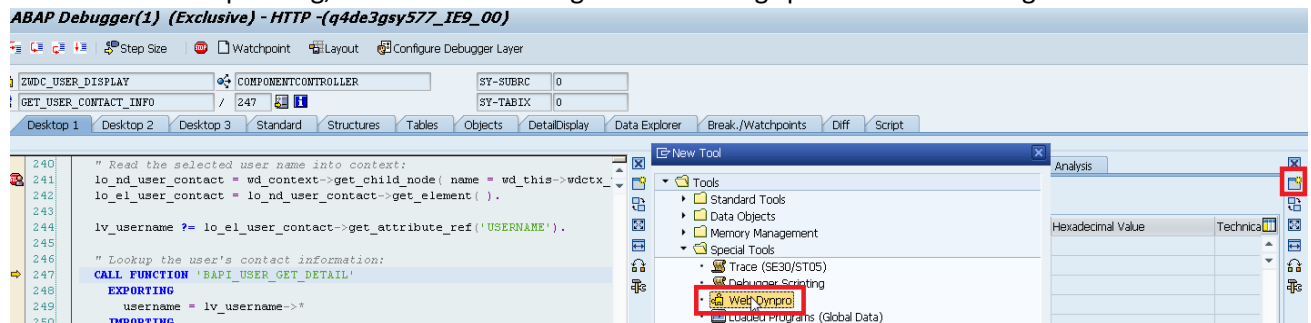
- Now, execute the WDA:

\*User:

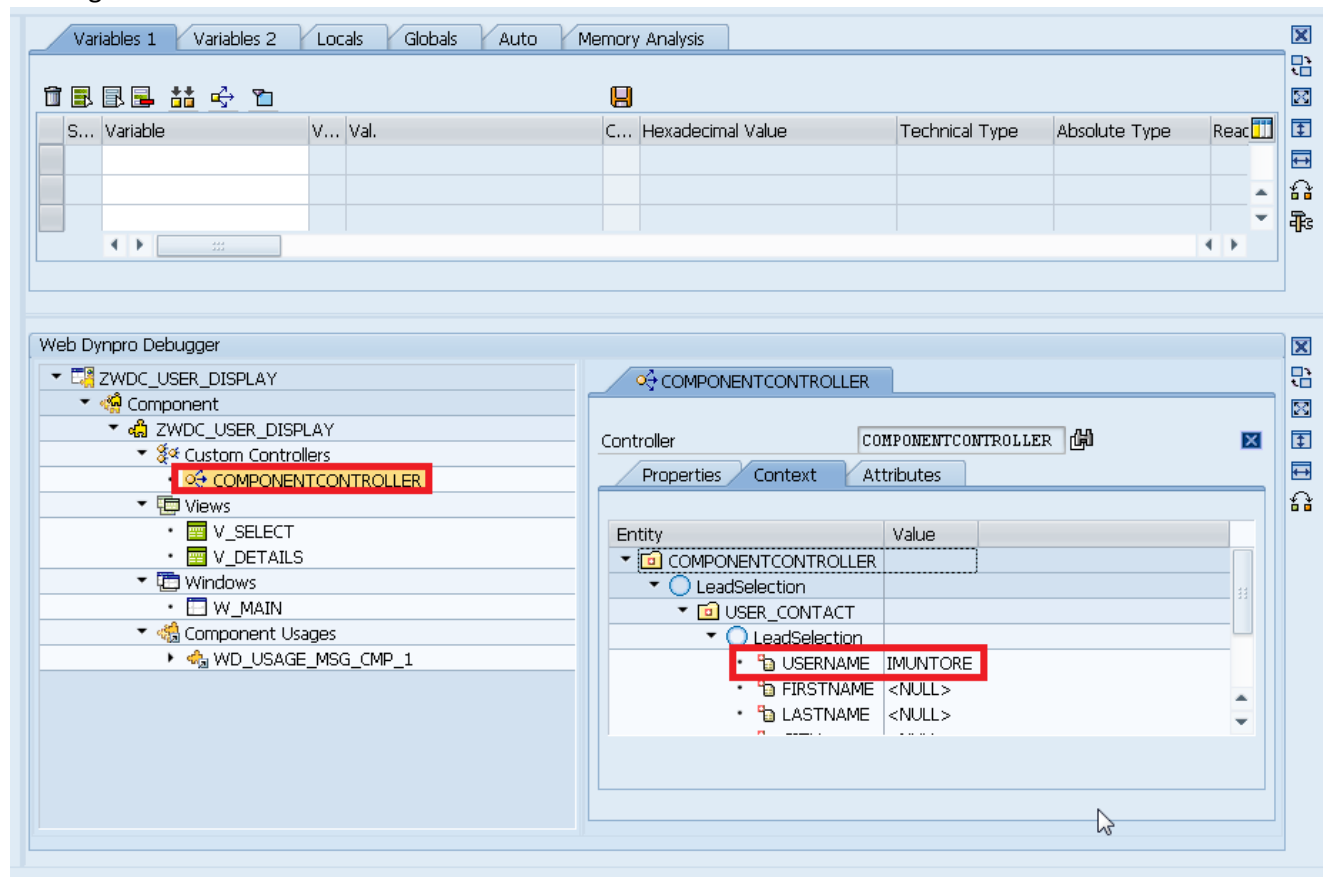
c. The debug opens automatically:



Go ahead with F6 pressing, until the BAPI calling. Call the debug special tool for Debug:



Your screen will be like the SS showed below. It is a very powerful that helps a lot during problems investigation:



**14) Homework:**

- a. Create a button in the V\_DETAILS to return to the V\_SELECT view.
- b. Create a validation if the entry data is valid. If not display a POP UP and freeze in the V\_SELECT view.