Audio file

[ABAP Webdynpro Training2 - 28042020.mp4](https://capgemini-my.sharepoint.com/personal/abhay_bhagwanjee-singh_capgemini_com/Documents/Transcribed%20Files/ABAP%20Webdynpro%20Training2%20-%2028042020.mp4)

Transcript

Is ready, input field is ready.

The system will know that this input field is.

For my vibe K field, it should hold the value of EBC.

So I should map this input field with the UM.

Attribute called verbmobil so which is under the node sales under score under score order so I will be mapping it with blme field so so it'll go to the layouts.

Select this particular field.

Now everything will be operated from the properties section.

So when you click on this input field, check the properties.

You will find that somewhere you have to do binding so it's.

There's a property called value.

Value means which value this input field is going to hold.

Now we have there's a.

You can see there's a highlighted button.

Here like when they click on this button you have you will be nothing but binding the property of this input field with the context attribute.

So now after clicking on this button you can see there is a context binding.

A popup has appeared where it is showing me what are nodes present in my view context and that view context node has these many attributes.

So I have.

I have to select that this input field should have.

Should be binded with which particular attribute so it should bind with blme.

We can see now the binding has done. It has binded our main view with sales under score order dot.

Field so the binding is done.

So this is my sales order label and input field and the binding is ready.

Save this.

Similarly, now we have only created the.

UI layout for our first field VBVBL and now we have a few 3 three more fields which is date, time and name.

So we have to design the similar kind of layout for these three fields as.

Well, now what I can do is.

I can again right click insert element or I can just copy the the label.

So I can do a copy element and I can insert that copied element in into my tray container.

So right click and do insert, insert, copy element.

As I have told earlier that every element should have a unique ID, so the system has automatically created a unique ID by adding under score. CP is a copy.

Uh in front of the uh the ID so we can give a meaningful name to this ID.

We can say that this is a date.

Now I'll be copping a.

So this will this is an input field.

Now the date will be.

I don't want the date to be an input field, I want it to be a uh just a display only field.

So there's one more UI element which I'm going to introduce now.

This is a text view.

So this this UI element will be only for display purpose.

It will not be an editable field either we can create an input field and make it read only so there's a property of readonly.

You can set it as readonly or you can create a text view if you don't want it to be.

Be editable in in your application, so let's keep it text view so we'll see how it how the difference between the input field and the text field.

Now the text view has a text property.

So same as the input field has a value property where we binded it with the context attribute saying the text view has a text property so we have to bind it with the the.

Context attribute.

Now we have to tell this label that this label is for this particular this particular UI element.

So I will go to the date label.

Let's see a label for selected and select V under score date.

Click on enter.

Change the text is because we have copied it from the sales or it has adapted the properties of that particular element.

So we need to make these necessary changes.

Make it as date.

I can see uh my tray.

In my tray I have selected the layout as a matrix layout.

Now they are these all elements are coming next to each other.

I want that the date should come at the in the next row.

So we'll select on the.

Date label.

And I'll say that this should come as in head up of my layout.

So in the in the layout data section you'll see there are two options, a matrix data and the matrix head data metrics.

Head data means this particular element will start from the beginning of the view.

So we will select on metrics head data and click on enter.

It will come at the beginning of the next row.

If I select this input field and assign the property as head then again it will go to the next line.

So this is how we can arrange the elements in the matrix using the matrix layout property.

So as all these elements are inside this row the.

Layout property needs to be assigned to the main container and then every element will have a.

The layout data property.

So the main container will have a layout property where we will be selecting that type of layout and every other element will have layout data property where we have to assign these sub properties of the layout.

I will now I will keep this input field in the first row.

So I will select the matrix data instead of head data.

Seats ready.

We will quickly add a time and date field in our layout.

So I will copy the elements.

Right click and.

Paste it.

So there is no paste option.

There is an insert copied element option.

I will make it as head data.

I want it to be in the next line.

Then this is.

This is a label.

Will change name as time.

Next now instead of creating text view for time, I will create an input field and we will make it readonly so.

Will create input field for time.

Find the value property.

And we will make it read only.

So you can see now it's not editable.

Will assign the value, sorry label property.

For the label of.

So this much is ready now.

Saved us.

So the last field was.

A name.

Now let's quickly add a name.

Hey, Cortana.

Hello, I'm Angie.

I I had one doubt like if we select the input field as readonly so if it's not editable so it will be like working as a normal text right?

So there is no difference between a normal text view when input field as readonly.

Right.

So input field has a certain limitation like we usually go for text view when we have a large text, we need to create a large text area in the layout.

Like we can define the number of we can define the width of the text.

OK.

So let me try to do that.

Other than that one more thing is sometimes you require a text field to be visible for some scenario or editable for some scenario and non editable for something I do.

So at that time this display property comes into picture.

So this text view.

Uh has when you want to display a large text in in your screen.

At the time we go for a text view and here is the property from where you can define the width of your text view.

So it can be.

It gives you a large work area for your long text to be displayed and similarly if you want to enter a large text in your view then you can go for a text edit instead of text view.

So text edit is also one of the UI element.

So I'll keep it short for now.

OK.

OK. Thank you.

Hi, my sister.

Actually in the node there we have the four fields.

So can we directly use that for field as a input into the view?

So in this view right now we are creating the four input field and that after we are going to assign the node field into that the input field.

So can we directly use the node field as a input field?

No, we can't directly assign the entire node to your.

Are you asking if I could select this entire node in this text, right?

No, no no. Right now in the text we are going we are assigning the 11 by 1 field right?

But why can't we use directly this node field as input field?

Like in the view you have created the text node or text uh.

Text fi.

Text field correct?

So first you you created the text field then for that text field you are assigning the code field node node field correct?

Instead of creating the text.

Uh, we can directly use the node field.

In the.

So basically what you're saying directly create create all the fields using the node.

Yes, yes.

What you losing the you.

OK, so that we can do so right now, while we're doing one by one, is to check the different properties of UI elements.

There's one option.

Uh, wherein we can directly right click on on the root UI element container I guess.

Uh, once you can try that.

Yeah, I'm trying.

Right click on.

Create container from.

Then create container form.

Then select the context node.

Like this?

Read the note.

And then define the.

UI element.

Defend it, yeah.

What kind of UI?

Oh yeah.

And then if you do this then all of the field will be visible there.

But normally when we do 1 by 1.

So our.

Approach is to.

Show the complete design one by one and to check on the properties.

So we are following that.

Yeah, so there's one option to go by this approach.

OK.

But it can't be that.

So I select.

Yeah, but it can't be like you have designed the form and now you want to map all the fields at once.

That you can't do.

OK.

But here is option like if you select the.

Uh that particular node under that I want to only select 2 field.

So can we do that?

Like here is the 4 field but I want to display only two fill.

In the view.

That then I guess you have to control through the visibility property because here we have the reference to the node so it will basically show me all the fields.

So once the form is created then you can select the property of the visibility and.

Only the relevant field will be displayed.

OK, fine.

OK. Thank you.

OK, you can continue with your design.

Uh, so we have now binded all the four attributes of our of node.

Let's activate this application.

Now my requirement is to select some.

A sales order number and on click on some button I have to do.

Play this data.

So what I will do, I have to write some piece of code behind it to get the data from the relevant table and map it to the fields.

So every view has a associated controller where we have.

Uh, these many methods are available.

These are called the hook methods.

So my view has six hooks hook methods.

If you read the text of this method you'll understand that what's the purpose of every method like WD after action if I click on some button I want to perform or after navigating like I want to do some action then after doing some action if you want to perform some logic.

Then you will be writing it in the after action.

If you before performing any action you have to do do any validation of the inputs.

To raise some error if the data entered is not correct.

So it will do it before performing any action.

Do exit is when you will exit from this view.

So it can be when you are exiting from the view one and navigating to view 2 so it can be like that.

Doing it is when you first entering into your main view.

So during the initially while these particular views instantiate it at that time this doing it is called.

Modify view is executed every time you perform any action on your view.

Even when you refresh your view, when you click on some button in your view or you select any value from the drop down so on doing any every single action on your view this the entire main view has been refreshed.

And it is modified, so modified view is called.

Uh and uh, when we modify the context of the view, the context menu on context menu method has been called.

So now as we saw that our purpose is to enter the sales order number and.

And we have to get we have to display the data from the table so.

What we will do, we will add a button so that after entering the number we'll click on that button and on the button our page will refresh and the other section of the layout will be populated with the with data.

So I will quickly add 1 button.

Now I I will I want this button to be by the side of this input field.

I will first give the text to my button get data.

I will right click here.

And click on the app.

So by clicking on app it will move up in the upward direction or I can just drag this and drop it into the place where I want to place it.

Now it's ready here.

Can see this button is a little Gray out because I can't.

There's no action property assigned to this particular button.

So if we check this button properties at the bottom you will find there's an event.

So on action event is associated with the button so and we will create any.

Uh, action on this particular button.

This will become.

This button will become active.

So let's create one action on button.

So this this is the name of my action, so let's continue.

I can see it has it has become active now double click on an action button.

You can see.

This method is ready, so it has navigated to a method column.

You can find that this is an action, so it has prefixed with an action.

Text so you can find this particular action in the action column as well.

So this is my action and then you will see in the in the method section it will be prefixed with an action to indicate that it's a.

A type of action, not an actual method.

So now I'm into my on action get sales detail method.

Here comes the coding part where we will be reading the input field data from the screen and using that data we will be running a query on DIC table and getting the relevant data and mapping that data with the other fields on the.

Uh, on our screen.

So first thing is we have to read the data, put a select query, then set the data on the screen.

So these are the three actions we need to perform to display the sales.

Data so how I will how I will be reading this this field data from the screen so as we have yesterday we have discussed why.

Uh, going through the PPT that web dynpro has?

Uh, less coding and more of a code.

Code wizard oriented programming, so there's one web dynpro code wizard at the top.

You can see it here.

So using this wizard, most of the code is generated.

So the reading the data from the screen and setting the data into the screen will be performed using the code wizard.

All you have to write is right.

One select query to get the data from the relevant table.

So I will here demonstrate how to use a code wizard so we will click on this particular wizard.

We can see there's a read set append as a table operation given by uh, code wizard.

Read is what you will be reading the data from the screen setting is you will be setting the data up and you'll be appending the data into a table.

So it's a table operation, so.

So for now we the our node is a kind of a structure.

It's like a work area.

We will be reading one single value and setting the values of other attributes.

Now I will perform a read operation, so we'll select the read radio.

Button which which attribute I have to read from my context is what I have to select from the context.

So we said that yes, we have to read now which context?

I have to read any two DBL.

So we'll double click on DBL.

And I will click on continue.

So what will happen it will read the value of rebellion which is entered by the user on screen.

So we will continue.

You can see the code is generated.

So few declaration is done for a sales order node.

This is our attribute within a sales order node.

So this is declared.

Then a sales order node has been instantiated here and this is a get element.

Property which will get the data of verbmobil.

So it will it will create a.

It will create an instance of the the node sales order and then it will help to get a data of an attribute called bull in the variable LV under score. So when we will debug will understand in VLC this.

This variable will be holding the value which will be entered by a user on the UI screen.

So now we have got the value of bleugh.

So what we will do we will put a query on our sales header table.

So we will be reading the date, time and name from this particular table.

Now let me.

As we know we have to put the the data which is selected into a node which we have already defined here so.

We will be putting the data into the context node which is LS sales order so.

Now what we have to do, we have selected the data from VBAK table and we put it into a sales order structure now.

Uh, we have, as we have done, our get attribute.

A selection of we have done a get attribute selection of rebellion here.

Similarly, we have to set the attributes after loading it into our sales order structure, so we'll again use a wizard.

Go to the set property.

And select the sales order node because I have to set all these properties.

Either I can set all the data one by one, I can I can set it all together.

So whenever we use a code wizard, it will give you the declarations and the code the node instantiation logic together all together.

But we have already done a get request before, so these declarations.

We're already done before, so we can delete this repetitive declaration.

You can see this LO&D sales order Lol sales order. This is repeated again so I can delete this LS.

Sales order is also declared earlier so I can remove this.

Even this particular node was instantiated at this line so I can remove.

This piece of code.

Now the get element was done here.

So I can avoid this, all I need is.

This set attribute property.

Now the data is in the sales order structure.

All you have to do is set the the data into the attribute.

So code is ready.

Uh, what we did is we we read the webcam from screen, we derive the other bits from the table and we set it to the other attributes on screen.

So now let's test our application.

So to test an application the first thing we need to create is a web link through application.

Nothing which we have created now is an component.

So to execute a component we need to create an application first.

So if we can right click on the component.

So right click on the component, go to create and create a web dynpro application.

Save this.

Activate all component together.

Component is active and then application is ready.

So the application name will be same as your a component name then.

So from the execute button you can execute the application or a URL is generated for this particular application.

You can copy these URL and paste it into a browser and you can check it.

So I can I will directly execute it from here.

So now you can see this is the view which we have designed.

This was our input field date was a text view.

So when when this date will be loaded with some data then you can see.

We will be able to.

See the the data in this blank area.

Time was an input field but in a read only with a read only property set.

So that's why it looks like an input field.

But we can't edit it because it is readonly and name was also a text view, so there's no boundaries for this particular text field, just doesn't look similar to the input field.

Now you can see there's a circle for sales order number.

How the search has appeared in here?

Because we haven't designed any search and so yesterday if you remember in the component controller section we have checked out few properties of nodes and attributes.

If you click on the BLM field we'll see that the input field mode.

Is automatic.

That means the in the DIC table there was already a search help associated to this particular field, so.

Uh, the automatic property will adapt the property which is there in the DC or we can explicitly add the DC search help name.

And these are the other two options for search help where we design our own custom search helps so fourvel.

Has adapted the property from the DIC.

Now let's select some sales order number.

And click on get data I can see that for sales order sales document #9 there was only date available. There was no time and name. That's why only date. You can see it here.

Let's find some good data where you can see all the three fields.

Yeah so you can see here name name has appeared even the date has been refreshed and there was no time in that particular table for this document number.

So this is how the data in the the these attributes have been refreshed.

Uh, so.

Let's quickly go to the application.

Go to the main view.

Put a breakpoint here.

So to debug an application we should put the external breakpoint and.

Click on get data.

So you can see we are now in our get sales detail action.

Go 1 by 1.

So now it is going to read the web.

They learn from the screen. So this is 356 document number which we have entered.

No, this is our basic select query.

This is our sales order structure which is populated with the data and it is going to get binded with the attributes in sales order node.

So this is how an application works.

Uh so now on the on the button click we have written an action.

If I if I don't want a button I want that after entering the sales order on click of enter these these elements or the data in this particular attributes should get refreshed.

So there is an on enter property for an input.

Or uh, when when we select anything from the circle at that time we can see the breakpoint has been.

Uh, your application went into debugging.

That means your page is getting refreshed at that time.

You can put your code in a modify view as well because every action which we perform on our screen triggers a modify view.

A method so we can put most of the logic in.

Modify view method as well.

So any questions till now?

Did you understand how can we design our basic layout?

If this is clear then we will move towards the use of inbound and outbound plug.

Yeah, hello.

Yes, we are.

When I come here.

Actually I have one question actually what is the difference between that automatic and by dictionary?

The moment we have showed the search help things right where we have one automatic and the search from dictionary.

So is there any?

Difference between both of them or is it?

So automatic means if there is any search help associated to that field in your table so.

If for this particular table for this particular field, there is already a check table associated with it, so it will adapt the property from the table itself and it will show you the search help.

Uh, by adapting the property from that particular table that that is what automatic is and if you have designed your own custom search help.

Some new DIC search L for this particular verb Glenfield.

Then you can go and select the DIC search help option and can click on enter OK give the name of that particular Z search L in the DHC search help.

A section so so when you don't want the field to adapt the property of the table.

A table element?

Then you can go for your own DIC searching.

Or you can just deactivate if you don't want any search help, so automatic will.

Uh, will.

Adapt the properties from the table, field and DIC.

A dictionary search L will give you an option to give any search L that you have created, if any.

Existing it can be any standard SCP search help or it can be any custom search shelf which you have designed.

One question from my.

Side that how we can change the field property dynamically.

A field property dynamically?

You mean like?

Visible, invisible or something like that based on.

Some value.

Yes, yeah, we can change.

We can do that.

Let me show give you a quick example on it.

So let's say uh.

Uh for currently however, your looks like.

When we don't have any data still we are getting this blank field.

So what I want when initially this field this page is loaded, I want these three fields to be invisible and when we get the data this should become visible.

So if we want to handle.

The visibility dynamically based on this particular button click and what we will be doing is.

So let's select one field.

In the properties section will find the visibility property.

If I select the visibility property from here then this, this, this this will not be handled dynamically from the UI.

It like if I have set it none then it will be always.

Invisible to the uh at your UI level?

At your browser, uh.

So what we want to do, we want to handle it dynamically.

Let's do it. Uh.

We will what we will do, we will create one.

We will call it as a UI property node.

Now this is node which I'm creating at view level because I want to handle the properties of this particular view so I don't have to create this.

Now at the don't have to create this node at the controller level.

Why should we put a burden on controller if we?

I want to do this stuff at the view level.

So we will be doing it at view level.

Now set the visibility property.

For DIY elements, now there's one data element for setting the visibility.

School WD.

This is a web dynpro visibility data element for setting the visibility of elements.

So we will find it has two possible values 01 is for none and 02 is for visible. So we will create 1 variable and we will assign the data element of WD UI visibility type.

Go to this.

So so this is our attribute ready with the WD UI visibility data element.

Now go to layouts.

Now you let's say I have to make this name invisible dynamically.

So I will go to this name.

Check the visibility property and from the binding option create binding option.

I will bind it with.

I will bind the visible visibility property of the name with this visible UI attribute which I have created.

So you can see now this visibility property is binded with that particular attribute.

Activate this.

Will go to methods because we have just now only binded that particular variable with visibility property.

Uhm, no.

Context, but there's no logic return to set it as visible or or not.

Now what I want when I will initially when I will refresh the page so the first view which I will get all of my.

Application at that time that name field should be invisible.

So I want in the initial stage in the while doing the initialization I want it to be invisible.

Now I will use this code wizard.

This set the property of visible UI element.

So to make it invisible, the possible value is 01.

And to make it visible it is 02.

Now we want that initially it should be invisible so we will assign a fixed value as 01.

This is now we are setting the property.

Can see there's a set under score attribute, so we have to give some value and say 01 which means it will be none.

Let's activate this.

Now we have make it invisible.

I will also assign it to the label so that it will be.

More clear.

While demonstrating.

Now we'll refresh our application.

Just remove this breakpoint.

We can see that the name label and the input field or the text view of that particular name is not visible.

So now I wonder if on click on the get data button it should become visible.

So we have not written any code to make it visible.

The property will be always set to 01, so we'll go to the methods. Now we will open the action.

So once the data is set.

I want that that section should also become visible again.

Using the wizard we will set the property of visible UI element.

And now we will set it to 02.

Now let's refresh our application.

Get some.

Document number.

Click on get data and see now this has become visible.

So this is how you can dynamically handle the properties of UI elements.

OK. Thank you.

Now let's move towards the inbound and outbound plugs.

When the inbound and outbound plugs will came into picture when you have more than one views in your design.

So currently we were only dealing with one single view, so there was no necessity of the plugs, but now as we.

Uh so now as we will be creating uh one more view and I want that on click of the button on 1st view I want to navigate to the second view.

So in this case I will need the plugs.

So let's now create another view.

So I'll click on the views, right click on the views and click on create.

So the detail view has been created again it has a transparent container or GUI element.

We will be adding our UI elements inside this root element.

So let's do one thing.

A query.

We quickly add few elements because now the elements demonstration is done, we have to focus on the navigation so it will not take much time to add elements.

So as you can see there is no context available because I have not added that context in my view.

So the first thing we have to do is to map the context from the controller in my view.

To activate it.

No, no, the dragging was not successfully bunch.

OK.

So for this tree I will be selecting a grid layout instead of a matrix layout.

Can you see in grid layout I can define how many number of columns I want to keep.

So I want only two columns.

I can see the all elements are arranged in in the two column pattern.

So that's the.

Advantage of grid layout.

So now what I want is I will.

I will enter a sales document number from my phone.

First view and onclick of get data I will navigate towards the detail view.

So instead of showing data in this bottom section we will go to the second view and check the data at the next in the next view.

So what we have to do, we have to move from Main view to detail view.

That means we have to go out of Main view.

That means we will be defining an outbound plug in the main view.

So we are we are defining an outbound plug, so I'll follow a convention like Obi under score and we are traveling to the detail view so I'll say two detail.

OK.

Save this.

Now while going from Main view to detail view I have to pass the BLVE which I have entered in my main view that things while navigating it should also carry the VB, VB, VB.

Dillon Field your sales document field value from Main View to detail view.

So for this purpose we need to assign the importing parameters to your outbound plug.

Seth Ivide LM.

And the type for web element is.

Hi, ashna.

So we have defined now the outbound plugs.

Oh, yes, Sir.

For the main view, now as we will be entering into the detail view, that means we will be defining an inbound plug to a detail view.

We'll call it as IB from Main View.

Save this.

It will be.

In the outbound outbound plug or in one plug?

Showing it in the outbound plugged right, but I think you told that it should be inbound.

Further detail view.

Yes, for detail view it will be an inbound plug because we are entering into the detail view.

Yes indeed.

And we are exiting the main view.

So why I mentioned it in incorrect?

I'm so sorry.

Yes, yes.

So that is what I want.

Yeah, yes, I.

OK. Thank you.

Thanks for writing.

So we defined a inbound plug for a detail view and an outbound plug for main view.

So let's activate our component.

So once the plug plugs are ready, now we have to link these two plugs.

How the system will know that?

Yeah I have to link my outbound plug of main view with the inbound plug of detail view.

So this linking of the plug the navigation link between these plugs.

Are defined at the window level, so Windows should know how many views it is carrying and from which view to which view we have to travel, so we'll open the window.

Can see for now there's only one view in my window because it was the default view generated while creating a component.

You have to manually add the other view.

So I'll right click on my window structure and embed our view.

Detail views the new view which we have created.

Click on continue.

So then my second view is ready.

So as I've told you yesterday that the the one which is highlighted in yellow color is the default view, it will be visible first when you will load the application and this is the normal view.

Now if we open this you can see there are these plugs associated with this.

So I want to go off.

I want to map these outbound plug with the inbound plug of my detail view.

So what I will do I will right click on my outbound plug.

I will create a navigation link.

Detail view.

So you can see now you have a right click on the outbound plug of Main view.

It is linking it with the inbound plug of detail view.

So the navigation has been set up between these two views.

Now our window knows that yeah he has to carry these many views and he, the the user will be navigating from this particular view to this view.

But code wise we don't know on what action we have to do this navigation, on what user will, what kind of interaction user will do and at what time we have to perform this navigation.

So we have to go to the main view.

So in our example we have we're we're considering that on enter of a document number, after clicking on the get data button, we need to travel to the next view.

So we know on this particular action we have to perform this navigation so we have to go to the action of get data button.

And oh.

So let me do try it.

At the bottom we will use the code wizard.

So apart from this context area, we have a general tab in the code wizard.

So there are various different operations given in the general tab.

You can explore this one by one.

Few of them are used to call the method, a few is used to generate a message, message, pop up, up.

And this one to do a navigation, so we'll be using the one to do a navigation.

So it says start navigation.

To select the plug.

Now you can see we are traveling from Main view to the detail view and also we are carrying the.

While traveling from Main View to detail view So what we will do?

The code was already written to get the web.

Len value will pass it to the.

So now we have put a logic saying that on the button of get data button action I have to travel from Main view to a detail view and along with it I have to carry the data of blehhh which is entered in the main view now have.

I will come into the detail view.

So when we define an inbound plug to any view, there's a method created which is called handle inbound plug.

So it will be handling the logic while doing the navigation.

Like as as we did the navigation from Maine to detail and we are carrying a Webgl from Maine to the detail view.

So handle IB inbound from Maine will give us a opportunity to read that input from the.

So this is a manual test to define the input field.

Score V8.

Now the data which is transferred from the main view, we can get it in the our input field.

So using this input field we can put our select query on on the table and we can set the.

So the attributes of a detail view.

What we will do, we will put a similar kind of query.

Now we have VBN.

Which is given by the main view.

We are using it to get the other bits from the table and now we will be setting the.

The data in the context of detail view go to the context set context sales order.

So let's see that this works or not.

Now we will refresh our application.

Select data.

So now we have entered into a detail view.

So earlier it was a sales order view and now it is a detail view.

So this is how we have navigated.

You can see the document number is carried from the main view and the detail view has done all the logic to determine the other bits from the table using the sales document number.

And then it has set the attributes to the node of a detail view.

So this is how we have navigated from main view to detail view.

Now I want to go back to the main view.

So how I will go back to the main views I I will need some again up button to perform an action and on that action I will use the plugs to travel from detail view to Maine.

For now we have only created a route from Main View to detail view but we have not created any view back from detail to Main View so.

That that is missing.

Second, everyone, please don't eat.

Hello, yes.

Any questions?

If you have any question you can drop it into a message box.

So so we will now we will see how we can travel back from detail to Main View.

Because now we are stuck in the second view and I want to select the data from the main view and again display it into the second view.

So there should be some option to go back to the main screen.

So let's.

Define the plugs to go.

Back from detail, we have, we have.

We want to go out to the main view, so that means we will be defining an outbound plug to the detail view.

Traveling to mainview from detail view.

And they are coming in mainview from detail view.

So we will define an inbound plugin.

Main view.

Detail view.

So first step is done to define inbound outbound plugs.

The second step is to go to the window and assign the directions.

And find that few more plugs are added.

We have to go from.

From detail view 2 Main view, so to go out of detail views I will right click on the outbound plug of detail view will create a navigation to the main view.

This is my main view.

This is the inbound plug of Main view.

So the navigation is set.

You can see the navigation link here.

Save this.

Then the third step is, uh, giving.

The firing the navigation on some action.

So what we will do, we will create one back button.

Yeah so I'll select the outbound plug from this option available so it will automatically generate a code which we have done in the previous previously using code Wizard.

So what we did in the we.

Didn't know code wizard is.

We selected the code wizard, go went into the general section and do a navigation.

We put a navigation code using that start navigation option given in the general section, so if you want to avoid doing that, we can just simply select the plug.

From here so it will automatically generate code on this button action if you see this.

See now the fire to the main plug code has been generated.

Because we have selected the outbound flight from there and the other option was to go to the code wizard, go to the general section, click on start navigation.

Select the outbound plug.

And continue.

See same piece of code is generated.

So these are the two ways of putting our code.

Delete this.

Let's go back to the browser.

Refresh this.

Daddy to 263 tools.

So now we are into the second view, will click on back.

Now we are back to the main view.

So this is how we can navigate from one view to the other view using inbound outbound plugs.

Any questions still here?

Manoj, you have some query.

Can you hear me?

Yes, I can hear you.

There isn't.

Mom, can you hear here?

Yes, yeah, let me tell.

Voices speak.

So one other question was that we have declared the parameter while declaring the plugs.

So he's asking how can how are we using that?

So if you see while triggering the the plugs from first view to 2nd view we are passing the VLM.

OK.

OK.

So now in the second.

We will in the second view we are receiving it.

So the second view.

And in the inbound plug handler.

There is a handler method generated for this where we are receiving it.

So here we have the LM.

So if you see we have written a query to fetch the data based on this sales order number.

So this is how we are passing the data between 2 views using a plug.

OK.

Using cabbages.

Can you be a bit louder?

Your voice?

Is too low.

Cables sending and receiving.

In better signal.

So it's basically an action, so.

In the handler we have.

There is no object directly involved.

In the inbound.

Plug we have the receiving part right.

Yeah, So what exactly you want?

We don't have option.

We don't have.

OK, in we don't have an option to receive the parameters in inbound.

Are you saying that?

We have.

Here we can receive those parameters.

So when you define an inbound plug, this method handler method is generated automatically and here we can define all the importing parameters which are passed from the outbound plug.

So whichever variables you have passed from the main view outbound plug, that can be received in the inbound plug handler method.

OK.

So I guess he's asking that in the inbound plug, we don't have any declaration in that inbound Plug tab?

At the bottom, yeah, there's nothing.

Yeah, yeah.

So that we have in the method part.

In the method you have to declare those variables.

He wants it.

Uh, so.

If we need to pass any table in that case we need to create any table type in sea level then we need we need to use the same here right?

Do you remember the Cardinal?

Yes, one is 2.

Hey sorry one is to end in that case it will be one is to end.

Yeah, so that will be a table.

But yes, for the receiving part we should have some table type.

Yeah, because here we need to mention some associated type, so table type we need.

OK so in that in that case for the detail view or the method handler method should trigger 1st and there then the init method will be getting triggered right?

Yes, because the view won't be called unless the plug is triggered.

OK.

So in Web Pro basically you just have to visualize like what will be the flow and then you can work upon your design.

So everything matters on this whole architecture.

If you are well aware of the architecture then you can look for your design or while giving uh analysis for particular requirement you can say whether this is feasible or this is not feasible.

Something like that.

So architecture is what matters.

Yes, and one silly question that without using the button is there.

Any other way?

From application toolbar or some other way to navigate from one view to another without using buttons?

What does button do?

Apartment is just triggered any action yeah that I can understand but without using the button.

I am just asking that is there any such way?

Any other way to trigger any action without button?

So you had an answer.

So we just need an action.

So even enter is an action.

Say on the enter part you can write your code.

Yes, OK.

So for a code to be triggered we in web control we need an action.

Now it is up to you what action you require or which action you want to do a particular piece of code.

And I have one question here.

So here we were just reading the data.

So what I wanted to know is can we modify the database tables or update the database tables with our URL link?

Is that possible?

Yes we can modify the data like we we have seen this example with all these fields were editable if you if you change them and if let's say we create any button called modify and on click of that button we will receive the we will read.

OK.

Again the modified data so so whatever user has changed we will read it again and then accordingly we will update our database state.

So for that will we use our normal above code or there is option in the code reader side?

So to do the modifier perform modifier action we have to use a normal ABAP code and to read the screen.

A screen values the modified values from the screen.

We need to use the code wizard to do the read operation.

So code Wizard only helps you with the designing part of the web dynpro and may be calling some of the standard web dynpro specific or context specific methods or attributes.

It just helps you with the design.

Back end everything is available.

It can take you to a particular method, but then you have to.

Write your app code.

OK and so now to read the value from the screen how we can use the code wizard?

Because earlier we used to read the value from the data table database table we back table right?

So in this case.

How it will be done?

So we will use this code wizard.

Let's say we modified the values in the screen.

So what we will do, we will go to the context, we will read the and either we can read the entire node or we can read the attributes 1 by 1.

So it will give you the modified values from the screen.

So shall I give you an example?

We have already done that right for the.

Yeah, if possible, yeah, if possible.

For the action.

OK so if we create a new button, modify button, so we have to write the logic in that inside that new button right?

So what we will do in that new button action we will again perform the read operation.

So that read operation will read the the modified data from the screen and then you can use your normal app coding to modify the database tables.

OK.

So it will be better actually if we get the access to some practice server system and so that we can try and ask queries as well.

Yeah, we will.

Naveen, have you got an update?

Sorry, hey, in India.

Regarding the ID is, have you got any?

I spoke to.

Yeah, she said.

After the call.

We will discuss regarding.

A human.

Can we call standard transactions in web dynpro applications like create sales order, create purchase order?

Yes, we can do that.

Is it possible to show an example of it by tomorrow?

Uh, I'll find uh whether I have any such a demo example or not.

If possible I will try, but yes, it is possible to call US standard transactions.

OK.

But can you please?

I'm not sure.

Show that how the.

Change we can.

Register enter or function key as action.

So we have seen that.

We have seen that.

Can you please pardon?

Yeah, we have seen that how the button we can use as a action.

OK.

So that how entered or any function.

Yeah I was about to tell that yes if you see this input field and if you enter some data into this field and click on enter.

Gap on it?

We can use.

So this input field has a property called on enter property.

So when you will create any method for this on enter property so on click of enter button.

This method associated to this event will get triggered and you can perform your action like someone has asked can we do a navigation without creating a button?

So yes on the on enter action you can write a code of your navigation and you can perform actions on on enter.

So for a field you need to go to this particular section event section.

So whatever event is written there, only those actions are relevant.

So here it was on enter so we can handle on enter.

Like for label there is on change action so every different UI element will have different events associated with it.

So only you can perform those events.

Which are relevant for those UI element elements.

OK.

How can we navigate queen or two windows means like if I have a main window and after clicking on submit button I have to.

And then there was a.

Move to a.

Popup window.

So is it possible or using?

Yeah, we can create multiple windows when when we go, when we create a popup in.

Web Pro it every pop up screen is associated with a separate window.

So if if you if you try to do some exercises on pop up screens you you can find it on CP technical.

So there you'll find you'll see you can see that when you design a pop up.

It is has a separate window associated with it.

Thank you.

And one more question from my side, can we have a transaction code for our web dynpro? Like instead of coming into S8 and then exiting it, can we have a transaction code directly?

Yes, we can.

And but that is of no use because what it does is internally it triggers the browser only.

So the layout would be similar to.

It will open in your recipes UI only, but it would.

Be a like a browser thing.

You can't write your normal app code to access to fetch the screen field values.

So it will always open a web browser only.

No, no.

Within within UI, will it help us to get a screen US?

Yes it will open in UI window only.

But like uh when you have a normal report program screen so there in you can write like access the fields using the screen structure something like that.

But when you create a T code for Web dental application, you cannot access those screen fields as a screen structure.

It's basically it creates an environment which is similar to a browser one.

What are you doing there?

So that is a possible case, right?

Yes, even that.

You can easily get an example so that in that basically you set few parameters in the transaction code at the bottom you have parameters so there we set that.

That is mostly never used.

It's like.

We did that for some.

Practice there.

Yeah, yeah, yeah.

Because see, normally within our app we are having reports and all.

So instead of that, instead of creating a module to have a separate text section or window section, we can have a web dynpro as well, right?

If you want to use it within SRP, I'm telling.

For normal ELV thing see web dynpro serves the purpose for the browser. It was solution to give applications on a browser. So basically on the portal thing so.

It's not advisable to convert your LV reports and those things into web dentro.

Yes, for practice you can think of a solution.

OK, let me.

Design screen wherein I give report or table in a web.

But ideal requirement of a web dynpro application is when you want to display application on browser.

Very great.

So basically a person has access to portal and he wants everything at one place.

So at that time we.

Design a web application.

OK, OK.

And and what if like we want to provide any authentication for our link?

Is that possible from SFB side?

Don't do that authentication.

Uh, yeah.

So that is basically controlled in the portal environment only because it is like what we call it as a single sign on.

So the person who logins.

So based on his authorizations and those things, only specific applications are or specific pages are visible to him.

So there's no requirement for you to handle some authorization checks in your web application.

If a person is able to see a web dynpro component, then it means.

He has successfully reached to the particular.

Operation Check center.

So that is why we say it's basically linked to port.

OK. OK.

And when you're maybe triggering a web application using your some custom program, at that time you can check your authorizations before triggering the web done through component itself.

So while designing web dynpro, no need to focus on the authorization part.

So that you should ideally do before your triggering part.

Yeah, because actually this was an example from my client, so we've been to application only because whenever I try to trigger their or link in our CG desktop so it doesn't work it it only works in the client's remote desktop.

So that means there is some validation, right?

Yeah, but were you able to open the application and then it was not working or?

I I.

I was not able to open the link only in our system.

Yeah, yeah.

So that is basically the authorization is done beforehand.

So that isn't a job of a web developer.

No, that is from the basic side.

So like when you see for the UI-5, right?

Oh man.

It's based on authorization, even similar to that.

Web Pro is also based on that.

Uh, where is like a web page, right?

It's a page, means one way is 1 web page.

Oh yeah, you can see like window is A1 web page.

OK, can we use multiple views on a one page single page?

Yes, that is why I said a window is a web page view is a.

A set of elements.

Instead of.

Yeah, right. Right.

So you can have multiple views in one view.

OK.

So that you can practice.

There is an element called view container.

You can explore that element.

View container.

OK.

Then there was one question regarding the table type.

Like a person told that when we are passing that.

Data using a.

We have to define a table type.

Then maybe you can think of a solution wherein you want to access the particular table between 2 views.

But you don't want to create a table type, then you can think of like how you can develop.

Maybe tomorrow someone can come up with this solution.

So you want to share a data between 2 views, but you don't want to clear declare?

A table type.

OK, so that's it for now.

I'm actually do you have anything that needs to be covered tomorrow?

I guess there's a table element?

Table left, yes.

Other than that, anything else?

Oh no.

OK, so tomorrow maybe those who have the access to system, they can explore the.

Session till now and rest hopefully, maybe you can get an access by today.

Maya, please.

Uh, take an example of standard code if possible.

Standard component, right?

Yeah like create purchase order, change purchase order or any any standard code want to see in web application.

Yeah, so that's what the man she told that she see if there's an application.

Yeah, I will try to find it out.

Yeah, yeah.

Yeah, OK.

But, but mostly we again there also we should focus on.

Right.

Thinking of solution wherein we can use a copy.

Instead of calling the standard ones.

Robert E Lee.

OK. OK.

We'll see you there.

OK.

OK.

Thank you all for joining.

Thank you everyone.

Thank you.

Thank you.

Thank you, ma'am.

Thank you very much.

Thank you, Hank.