Audio file

[ABAP Webdynpro Training2 - 29042020 (2).mp4](https://capgemini-my.sharepoint.com/personal/abhay_bhagwanjee-singh_capgemini_com/Documents/Transcribed%20Files/ABAP%20Webdynpro%20Training2%20-%2029042020%20(2).mp4)

Transcript

So now uh, I will share my screen.

Yeah, you can continue.

OK.

So yesterday we have seen how to design A view and how to navigate between one view to the other view.

And and we have covered a few of the UI elements like transparent containers, tray input, field labels, buttons.

And today we will be looking at how to design.

A table and then I will show you an example where we can call a standard transaction codes from web dynpro and after covering these two topics I will hand it over to my for continuing the other advanced topics.

So just tell me.

Never mind.

Sorry to interrupt.

Naveen, can you stop the previous recording because it is already 30 minutes it is counted.

Can you start the new recording from now on? Because it is already 30 minutes, it is showing 31 minutes.

I started now only.

OK.

So I have one uh demo example for a table.

So I will let you will go through this particular example.

This is the one which I've created.

It is an example of navigation and also a table URL.

Uh, so yesterday we have seen how to create multiple views, how to navigate between one view to the other, how to create a link between a view.

So there are three important steps to to remember for doing a navigation from one view to other.

And those three steps were first defining plugs at the view level.

So let's say if I have want to go from Main view to airline details view then I will define an outbound plug in my main view.

So I will define an outbound plug at the main view and I want to come in the airline details view.

From main views I will define an inbound plug in airline details view.

After defining the plugs, the second step is to add.

The window level.

And include these views inside a window, so there will be a default main view always.

Uh, embedded inside a window and all the additional views apart from main views has to be manually added in the window component.

So the the main view was my default view.

I have added a airline detail view inside a window.

Now we have to set up a link between the plugs at the window level.

So from the main view I want to travel to the airline detail view.

So we right click on the outbound plug and create a navigation link.

So if I open this you can see there's a link.

Created between this outbound plug and the inbound plug of Main main view.

So this is the Step 2 and step three is uh.

Now we need a code where we will say that on this particular action we need to do this navigation activity.

So on main view we have even we can have a button on which on the on the button click we can do the navigation or on a action like interaction.

We can do a navigation so a piece of code is required to uh do a navigation between the views and that is handled by a wizard so.

We did on a button click.

Uh, was that button?

So by selecting button you can find the action associated to it.

So here you can see there's a fireplug code return, and to generate this code we use a code wizard.

From the general section we need regardin outbound plug from the start navigation option.

So these were the three points to remember to perform a navigation from one view to other view.

Now in this example in the second airline details.

You have created one table so.

Let's let me show you the table UI element.

So this is how our table looks like.

To create a table the the the one first thing we need to keep in mind is to assign the cardinality of a node.

So as I have told you yesterday that if a node has a cardinality of 0 to one or one is to one then that means that.

Node is going to act like a work area, but if we assign the new cardinality as zero to N.

Then it is going to act like a table.

So what I have did here for airline details, I have assigned the collection cardinality as zero to N that means this node can hold more than one entries, more than one entry.

So we have zero to N cardinality.

Set selection as zero to 1 because I want to select only one row so we'll focus on the collection cardinality because this is what going to define this node as a table.

So if we have set it as zero to N then what advantage we get in the?

At the code level, is a.

So this table is already ready.

I'll create one more table just to show the demo.

So this is my tray.

Inside this tray I have created this table and one button.

So below this button I will create a similar kind of table so.

So I'll right click here.

Insert an element.

Select to select an UI element called table.

I can see a blank single one single row and few columns.

Sorry one column and few rows.

Table has been created because this table has no idea that how many colony has to create.

But it will depend upon how many attributes are there inside my node.

Uh, so we need to bind this table with our node, then it is going to it will adapt the property or it will adapt the number of columns according to the number of attributes in my node.

So just right click on the table.

And do create binding, so we'll click on create binding.

After select the context that I have, I want to create a table for S flight under score detail node. So I will select this node.

I can see.

Uh, underneath this table, uh, these many columns are going to create and the column every column will have will have.

We'll have a the every cell of the column will adapt property of one of the UI elements.

So it it is karady is going to act like a text view.

I can make it as an input field or radio button.

I can change the property of a cell from the cell editor.

Of a table column.

So by default it will assign every cell as a text view UI element.

It's just right.

Go go continue.

And see now inside the table UI element.

Uh, uh.

All these.

Sub table columns have been created.

The number of columns are equivalent to the number of attributes in the R node, and then every table column has a label that that's a caption, and a cell which is a which is of type.

Text view we can change this the property of the cell.

Currently it's text view so you can it will be only a read only field.

You can right click on the text if you want to change the property of this.

So you can do a swap element definition.

Select some other element type.

Let's make it as input field.

Now you can see this field has become an input field instead of a text view.

So this is how we design our table.

Now the next thing is now we have only covered how to do binding of context with our table UI element.

How how we will be populating data in this particular table so.

Go to the methods.

Yeah, so here you can see.

In earlier examples we were doing select single and populating our node because it was a work area.

But now our node is a table, not a work area, so we are selecting multiple rows from our.

Table into an internal table and we are assigning this internal table to S flight.

S flight details node.

So how we will generate this binding table code?

So again, we're going to use a wizard.

Go to the context, I can see there is a table operation checkbox, so you have to set.

Air contact your node as a table operation.

Now the stable operation checkboxes enabled because the system knows that this particular node has the cardinality of 0 to N that means we can perform table operations on this node if, if, if we would have.

Set the cardinality as zero to 1, then this checkbox.

Will will have been grey grey out because then the Web Pro knows that this this is a this is a type of work area so we can't do a table.

So it all depends on the cardinality what operations you can perform on that particular node.

So in this example this is a of type table, so we can do table operations.

So we will do set.

We will perform a set operation because we want to load the data in this node and we want to load it.

As a table.

So I've checked the table operation.

So now you can see there's a bind table method has been called and it's importing parameters is an internal table.

So this is how so our purpose will be to populate this table table with the relevant data as I have done here and do the.

Binding so all the declarations are done.

Uh yeah, you will do the selection and put the data in ELT flight details table and this line of code will perform the job to set the data in the table node.

So this is how a table.

In web dynpro works.

So I will keep this code as it is.

So let's test this application.

To test an application I've told that yesterday that we have to create a web d'enfer application.

So what we do, we right click on our web dynpro component and create an application.

So this application is already ready, so I'll double click here and click on execute.

Let's select the airline code.

So now you can see this was the table which was earlier designed and this is the one which we just have designed.

So now this table has those many number of columns which are there in our load, the number of attributes which we are associated to a node and while designing a table.

The default property of the cell so the UI element associated to every cell was a text view.

So you can see this is these are their text, they are not editable.

But for this particular table we change the UI element property of this cell as a input field.

So you can see now this has become editable.

So this is how you can play around our table.

So here we can change the data and perform an update operation using a map logic.

Yeah, I have created a back button to go back to the initial screen and there's a clear button to clear the data.

So what this clear button will do, it will just update these two attributes data with blank values.

You can see this is how it works.

This was.

Yeah hi sheet here. So when we exude the table like details so by default I guess 5 #5 number of rows are coming.

So is it something standard or like we have to modify it?

So by default we can find the table properties in.

In the table properties Scroll down, you can see the visible row count is 5.

So you can change this row count to 10 or already anything or if you want to change it dynamically like like.

In the method I will get the data from my database table and then I will I have stored my data in my internal table and then I will read the number of entries in my internal table.

Let's say there there are 10 number of entries in the Internet.

But so I want that this table should.

Uh, should.

Dynamically get updates with 10 number of columns.

Instead of getting a scroll bar, I should get 10 columns because my internal table has 10 entries.

So at that time I have to perform.

I have to change this property dynamically based on how many number of entries are there in an internal table.

So in that case what we will do?

Instead of defining a constant value in this property, I will create 1 variable in the context that variable is of integer type and then I will assign that variable in the binding area.

So in the context we will be creating one variable.

Then you will be able to see that variable in your context list.

So we will assign that particular variable to your visible row count property and in the code when we will get the data in internal table we will calculate the number of rows in that table.

And then we set the property of that variable, we will set the value of that variable.

Uh, by the number of rows in an internal table.

So it is similar to what we have done yesterday to do set the visibility.

If you were there in yesterday's section, what we were handling the visibility dynamically by setting one variable here.

Yeah, yeah.

So same we will do it here.

We will create one variable, do a binding here and then set the value of that particular variable from your method instead of doing it.

And this and the code for this variable will be written just after we get the values for that table like the place where we did the set operation.

So below that we can write right?

So once you get uh get this uh data in the table and you have binded it, you can write a code.

A piece of code here.

You can read the number of entries in this table and then you can use the wizard to set the.

Variable value then the set operation will be performed.

You can assign the count to that particular variable and it will dynamically it will change the row count in in a table.

OK. OK.

OK.

Hello mercy if we are changing anything in table values then the corresponding node will be automatically updated with the updated value or do we need any some code or?

OK.

State no.

Uh if if I change any value?

If I change any value in my table like I have made it a zone input field so.

If I change anything here, so at the back end when you have to again perform a read operation to read the updated node value, this node will hold the changed data and what you have to do is you have to read this line item again to read the updated data.

To read the line item, means from.

Higher from the node, right?

From the note here.

OK.

And is it possible to make?

The input for a particular cell.

Yes, so currently.

Uh, we have.

Change the cell property for entire table so.

I think we can do it for one single cell as well.

Then we need to do it via coding instead of setting the properties from from the this section.

Yes, OK.

So basically there are some standard existing codes like for each element which we can refer there.

There are some standard demo applications from SCP as well wherein we can play with the table properties.

So there are some standard classes.

And methods for each of the UI element.

So that needs to be explored.

But yeah, those are feasible.

Uh, any questions on table?

Uh, shall I move ahead with our decoder demo?

So I hope this table UI element is clear.

Now I will give you a quick demonstration on how to call our transaction SCP standard transactions from weblink replication.

So this is one of the demo application.

So for in this demo I have called a sales order transaction code from my web dynpro application.

So what there's a simple design.

What I'm doing here is I'm asking for a sales document number and.

Then on click of.

For sales order button we are calling tcode for sales order and whatever number which whatever number we entered in the sales document number it's to be passed to the transaction code sales order field.

So to do this what we have done on this button click.

There's a there's an action on that button click where we are calling an external window which will open web or app GUI screen and it will load the T code for VB.

For sales order that is V02.

So first thing is we are reading the Webgl field value from the screen.

The input field which we have created then.

These are.

Some these are some server details which we need to get like what is the host and port number.

Uh, because while while running the transaction it should know at what port and at what host we have to run that particular transaction.

So this is a standard class which will give you the most number and the port number.

So you have to generate a link.

So this is the format of a link.

So this is how we call a transaction code.

So it's a HTTP colon, the port name, followed by the sorry host name, then followed by port name.

So so this is a.

The preceding link for any transaction then we have to tell the SIP client that we have to trigger a transaction via O2.

Uh, in via zero to screen, there's a field for sales order, so that field name is Vibeke hyphen VBL.

I have to populate this field with the sales document number which I have entered on my web in pro screen and this number is what I have get in the previous.

This number.

So either you can hard code any value. If let's say I want to open the document #2, I can just hardcode 2 but I want to make it dynamic.

So I have made it as an input field.

I'm reading that value and passing it in the URL.

And this is the screen code the the VA zero to initial screen OK code which IS/00.

So now we have created the URL, put into an variable and assign this URL to the create external window method.

This is.

This is going to trigger a new window and going to call this particular URL, and then there's a open method.

To open a particular window.

This will create a window.

This will help to open a window.

Now let me execute this.

So I've selected document sales document as file.

Let's call the sales order.

T code.

So we can see now this is my first page and it has triggered a new window in the browser.

So a new window is created for AA0 to transaction and it has opened using that open method.

I can see the order number is prepopulated with value 5 because we are passing this, we are reading this data and we are passing it to the URL so it has a.

It has auto populated these order field.

If we check the URL can see it's via 0 to transaction and VBA cavlc field has been populated with.

Uh UH-5 value.

So this is how we can call standard SFP transactions using Webmail pro.

So if you want to call any other T code then a few things we need to.

Remember, is a.

So say if you want to call any other transaction, I want to call a BP transaction, then I have to first check what T code I'm getting for that particular PPA transaction in the back end.

So you can open that transaction in a new section and you can debug that transaction and I can check what.

Decoders BY-T code. You are getting at the back end level for that particular decode. So this transaction should be assigned with the S5F in T code.

And after percent 20 you have to assign the variable name to which you have.

We want to populate some data like in this case it was sales order number let's say in BP if I want to pre populate the BP number then it.

Then we need to check what is the field name for that particular T code.

So himanshi, so in this case we use V02, so if we change the sales order data OK it can be like number of items or some material description so.

Uh, how it will be updated in the database table?

Like normally whenever we change the sales order within our recipe so it gets updated with the user ID right?

Like changed by this user ID?

Yeah, so in this case like on which uh name it will get changed.

So as you have logged into your web application using your user ID and even the.

The while when we execute an application so it logins to the portal, it's it.

And what if I directly use the URL link instead of going through the web web application?

If I directly go through the URL link, so that means I haven't provided any user ID and password, right?

So yeah, if you directly if I closed everything and directly enter.

Please copy this URL of an application and try to open it in a browser.

So before opening the application it will going to ask you a user.

ID and password.

OK. OK.

I was just giving a quick uh.

Example like.

Need to check the data.

I guess BP won't be configured here.

Business partner, those data might not be available.

Oh OK, just quickly checking that I can show then.

Uh, so.

So you can see here the screen OK code is BUS, under score, main under score, enter.

So this is let's say if you want the.

Uh, the page which I have opened on front on the BP transaction and click off enter that page to be loaded from your web N + K and then you have to capture the OK code for that particular screen and you have to replace the URL.

Steve Carell.

So I have to replace the the DYNPRO code by the OK code which you will be able to see in the debugging and the transaction code will be the T code for that particular.

Standard transaction and then the field which you want to prepopulate is what you have to check at the back end with what is the.

Field name so.

So I think to do more debugging.

So what you have to do here is to identify what's the field name for this particular business partner debugging and then identify the field name.

Replace the URL field name by that.

By that name and assign the appropriate value and perform the operation.

So this is how it's it's possible to call standard transactions from web dynpro.

So that's the only piece of code required to do to trigger our T code.

Any questions here?

Long time oh OK too.

To completely OK to complete my part, you can continue with your area.

Yes, thanks Marcie.

Thanks for sharing your knowledge.

Hi, hello mate.

Yeah, we're there.

Uh, actually I didn't get that. I didn't get how to generate the URL for if suppose you created for V802 and I want to create for me 20 million.

So the format of this year and this format of this URL the standard I have even I have get this from Google that this is how we call a decode.

How to generate URL?

All you have to do is you have to replace this transaction equals to buy.

Your T code whichever decode you want to call and if you want to pre populate any field in your T code then you have to replace this field name by the by the field of your uh.

Transaction code and then it will be equal to the value of that particular field and this is the screen OK code value.

Other other string will remain as it is only.

These are the variables like the transaction code, screen field name, field value and the dynpro code.

Rest you can copy as it is.

OK. OK. Thank you.

These days I see a link.

Uh, yes, I guess Mikey, this is SSCF link, right?

It has a link to that particular path basically in the system.

OK.

So if you open QQ you can see the same way it's a it points to a particular folder using this.

So basically what it does is it opens the app GUI in the browser.

It's nothing specific to a particular T code because if you see the code that is opened in.

Once you can open that if it's already.

So if you see you have the menu and everything. Now from here you can navigate to AC 38, me 22 and whichever T code.

Possible you can go there so it's not specific to a particular T code basically.

It just opens the environment of UI in a browser.

So basically somewhere CP has designed that that it will support UI in the browser and we are reusing that.

And I don't think such requirements come.

If it is, then there are.

Few specific ones maybe.

OK.

OK, once you can stop sharing.

Thank you.

OK.

Estimate I'll share my screen.

Everyone can see my screen.

Not yet.

Yes, we can see it now.

OK, so till now you are like well aware of the overall web dynpro architecture and where all we declare the nodes, attributes, how we use the elements and each of the tabs of the web dynpro ABAP.

So using that.

So as I mentioned yesterday that what you need to understand is what is the flow of a web demo application.

So from where it starts and if we had a particular screen, if we do any changes, what would be the flow of of?

Each of the window, view and component.

So that is primarily very much important to understand.

So I'll start with few things which are like some helpful things which you might get in real time and other than designing a particular component from scratch, there might be a case that you might have to work on the enhancements.

So mostly as I mentioned that web dynpro component.

So mostly or ideally useful with uh In Sync with the portal environment.

So in the portal we have web dynpro components and we.

Uh, work on that.

So such requirements are mostly related to HR.

So you mostly get HR applications and you have to modify or enhance.

Those things.

And for HR, SAP has extensively provided multiple.

Uh like standard components and those are mostly related to access that is employee self-service wherein which are related to like leave application, timecard adjustment some. There are then LMS module like learning management server or system whatever we call so for that.

We have some standard components provided by HR SFP, so I'll start with enhancement.

As of now, so the standard components provided by SAP for HR starts with HR, then we have.

For employee self-service we have s s.

So let's see.

Let's go with the timesheet that is.

Related to Katz?

OK.

So we let's open this particular.

So these components would mostly be there in your system.

So if you see these are the components.

So what happens are for these components there are no standard applications that such like application name is not there.

So what we have is this component is directly configured in the portal.

On the ivu that we call.

And whenever user logs in, there's a page where he can see multiple web dynpro components that he can access.

So one of this is like related to time shift.

So you might get a requirement that.

We have a.

Called calendar popup.

Let's say.

Work list.

So this window has view of V under score worklist, so we'll go into that.

OK, yes.

Not this one.

This I need a particular UI wherein we have some elements.

OK, so let's take an example of this.

So suppose you might get a requirement that this particular design is not much useful and we want some additional feature wherein a particular user clicks on something on a button and we want some other confirmation or something to be done.

So this and they don't want a new application because one option is you can copy this particular component and you can create your custom component and you can make any modification they require but that is not advisable since.

Maybe in next update you might get some additional features in the standard and due to your copy you won't be having that features.

So only option left is you need to enhance this web dynpro thing.

So how can we do and how can we enhance the web dynpro component so it's very much similar to our normal one.

So if you see there is a spiral.

Button here.

Which is an answer.

So you can click on this.

It will again similar to our existing.

So now if you see.

We have this insert element option enabled.

So we can add our element.

Or do any options that we want we can do that in the UI level?

Now this is one of the requirement and mostly uh.

You won't get these ideally, but you might get that.

Uh, there's a button, it performs some action and you need to perform some additional validation on that.

So what is?

What you can you will have to do is like let's see if there's a code on this.

Action on action OK.

OK, so you have this code.

Now what is the requirement is uh before these checks that are happening we need to perform some validation or maybe we need to update a particular log table wherein we are logging that when this action on action button was clicked and by whom and we are keeping a log of that in a particular custom.

So what you require is that before this action gets performed, we need to store the data.

Or maybe we need to store the data after this validation gets performed or you want to completely change this logic.

So if you see I have clicked on the announce button but there is no such option to write up my code here.

So what we can do is.

If you see.

When I created an enhance, what happened is we have got these three new columns.

That is pre exit post exit over at exit.

So if I open.

SCT window again without.

The announcement.

The screen you how will the application be visible?

So this is my screen.

My view was.

Let's see.

My view is recorded in.

I haven't clicked on the enhance button.

Let me see what happens with change button.

So change asks for access key, so we don't want to modify.

As you will miss the recipe support.

So if you see.

If you see.

There are no such columns for pre, post and those exits.

So what happens is when we click on enhance that feature gets enabled.

So let me again go to that window.

So what happens is in pre exit.

When I click on pre exit.

Or this announcement gets opened up and whatever code you will write in pre exit that will get called before that particular specific method.

When you click on post exit what happens is this particular logic gets called after that particular existing.

Piece of work.

It gets executed.

And overwrite.

Is it like overwriting that particular code?

So whatever you write here.

Will overwrite that particular logic.

So instead of calling the standard logic, it will call.

Only this particular logic.

OK.

These are the ideal scenarios wherein you can do announcement.

So everything gets opened and you can create your new attributes.

You may create your new.

Methods so when you click on announcement.

But ideally when we are doing we should.

Avoid these things.

Because it impacts during the HR upgrade or those things.

So you might get something some weird behavior due to your announcement and those things.

So only thing that suppose you're not getting any solution then you may reach to this particular pre, post or overwrite thing even if you do overwrite.

Try to see that you are reusing the existing logic and then you're making some modification.

And so like that and suppose you want to see like you have an component and you want to see that like you are getting a requirement that when I'm doing some particular action on web dynpro, something is up.

OK.

So you get the component name and now you want to see whether there's an enhancement or not.

So just open the component in sea T window and in the least this enhancement implementation would be there.

And with that you can get to know like.

OK, there was a custom announcement then let me check further.

There's something written in this particular announcement that is impacting my behavior in the portal.

OK.

Any question on this?

Yeah, uh, just a question.

Like, uh, if once there is an.

Upgrade so will this be?

Will these enhancements get overwritten or some other thing?

Will there be any impact?

No, they won't be overwritten.

We would get that in our.

Those list what we get as spouse PD there.

You can check on that.

Like you want to keep that or.

Those won't be overwritten.

OK.

OK.

So when you open this one technical detail, you can see like where the announcement was made, so you can see this.

This was the view that was impacted.

So you have seen like OK this is the component and I I can make this announcement but in the portal how will you come to know like OK this is my application but what is the background component that is used for this particular application.

So how will you get to know?

That so?

That is possible using.

So let's.

Open this particular thing.

So what will happen is in the portal functional will show you OK this is my screen and when I'm clicking on this some something is happening and I want a particular piece of code return.

On this particular action.

So he will not be having any knowledge of the web dynpro component or anything, he will just show you this particular screen.

So to get the component what you should do is you can do a right click.

On this? Don't know.

If it will work here.

Or it should work on the port.

Yeah so you will do right click and you can click.

On technical help.

So here you will get to know the component name.

The view name as well.

Then you can copy this and go to your system.

And then you can check on that particular application.

So if it was something related to in the technical name, you might get a component name, as is the cats, whatever it was.

Got it.

OK so mostly in the support projects this would be helpful.

Like to get the component name and check on the behavior like what is happening.

OK.

Now if you see this particular application.

Here you have these properties, OK.

Then you have one more tab which is parameters.

So what is the use of this parameter?

So as you saw when we executed.

Did I close the browser?

When I am executing this particular web dynpro application, I am getting some parameter here.

So it is telling me OK run this particular application.

So again someone asked about the ICF.

So if you open QQ and if you go to this particular path.

You would see the name of this particular application there as well.

So what it is telling me to run this particular application and it is passing this particular parameter that the language that we need to use is in.

So similarly you might get a requirement.

Uhm, let's say.

Let's say you have an A timesheet update and in the manager overview you get 10 screen wherein.

Uh, he gets to see the timesheet update and on the click of button he needs to approve or reject and each of the button will call a specific web dynpro component.

OK, it's a separate component.

1 component is to display the timesheet records.

Those are updated by uh.

Employee and on the click of button he will call a new component which is specific to approval of a particular timesheet.

So now what you have to pass is maybe the timesheet.

Uh, key fields of that particular timesheet entry to a second component.

So one way to do that is you can create a parameter in your component in your application basically.

So let's say uh.

OK.

So now.

Save it.

See if I can replace this.

OK.

And whatever you define here.

That you can access in the application at.

Window level, and there's a method for that.

Which is handle default.

There's a piece of code that we need to write in the handle default to access the parameters.

Let me see if I can just write the parameter and access it.

I don't have a ready example for this, but let me show how it works so I'm refreshing this particular application.

So now you see when I passed Ivy data under score here.

We can I can read that in handle default method.

And then support.

Suppose I want to do some logic based on this particular parameter, then in the handle default what I can do is I can set this particular thing to a particular node at component level.

From where I can access those parameter in whole of my?

Application and then I can perform my logic.

Uh, did you get this?

This is one of the trick that.

Might be useful to you.

Any questions on this?

So by using this method can we change the language of the web page which we are getting?

Language order.

Web page.

Like anyway.

No, the.

T code so default language is English no.

So is that option available by using the method section to code it to get a fixed language of German or French?

Something like that?

That you can pass when you are triggering the application.

Those are actually standard parameters to set the languages and all.

So that is not the main purpose of this.

OK.

So the main purpose of this particular feature is to pass some common data or some key data between applications.

So between windows of a same component you are passing the data, maybe using plugs or maybe declaring.

Node at component level but there might be scenario that you want to pass data between 2 components.

So at that time this parameter would be helpful.

So as of now you have declarators K2. So can we declare the structure also here?

And also.

Yeah, you can do anything.

It's just you have to populate that.

Uh, not the structure I think because you can't pass that in structure.

It should be a single feed.

The length can be high.

You can't pass structure here.

Because it would be similar to this URL.

Like you will pass the value using the equal to value, so it can't be a.

Structure because.

The URL doesn't support that.

And the the requirement wherein you want to modify the language.

So in that case what will happen is you will have to create the URL.

Uh, you have to create this particular URL.

This thing and then you will have to modify the language equal to German or whatever you want.

OK.

But understand the purpose of these things, right?

Yeah, I got it.

Because our main focus is like what all things are possible in Web app.

Then you may explore on various requirements and maybe when you are getting a particular requirement then OK you might have a click OK, this thing is possible and then you can explore how I can do that and.

What other things are feasible?

So only one parameter is possible or multiple.

No, no, no, no. See.

Multiple parameters can be passed.

Uh, I don't exactly remember.

Like we have to do comma or well, it's similar to this one, so we'll refer this.

The button click.

OK, so we have to use this semi colon.

Both are very much similar.

So the parameters of a windows can be used as parameters of outbound plug.

Switching between users.

In case of like or passing any data within components, so the parameters of Windows section can be used correct?

Basically when you have this.

This parameter value you are getting from the application level and it is passed to the window.

Got it.

Main window, the default window basically of that, but now you need to set that at a particular.

Position from where?

Either you may pass that using a plug, or maybe you can define at a component level to be accessible to multiple views or multiple windows.

OK.

Then there are some of the applications web dynpro component called as faceless component.

So what happens in a faceless component is basically the term faceless means it the view will not have any UI element.

So the requirement is you are on a web dynpro page.

I will take the same example that I am clicking on approve button and it calls a new component to approve that particular thing.

But my requirement is I don't want to display any UI element.

It should just happen that it happens in the kind of background.

Or that thing?

So in that case, what happens is.

At that particular moment the use of these parameters is very much helpful because the second component which gets called on the approve button, it will not have any UI element.

So the data gets from the gets passed from the first component needs to be reused and.

We need to perform our action and then it's complete.

Nothing will be visible.

So at that time what happens is we pass the parameter, we use that.

Maybe at the window level we perform our action of approval.

Code that gets triggered maybe in the unit unit level?

You can after the handle default gets called in the init code you can use that.

And perform your whatever logic you need to do and nothing gets opened up.

OK, so how to handle the window open or close that you can do using.

Using this particular component.

So this particular component helps you to perform the actions related to window.

So in the init you can perform your action and then you can call instead of open you can call window close.

OK, so you just need to get the window manager details and we can close the window.

You can get standard code for this one.

So what things you can do on a window are.

In the interface you have.

These particular things that you can do.

OK.

Oh no.

So that was at the application level.

So when you create a component.

So till now what we were doing is we were.

Declaring a node at component level then.

Binding that node at view level, doing or manipulation, displaying data, navigating to views using the data component level.

So your code was written maybe at view level or some of the cases at the component level.

You can write component control level, you can write your code.

What happens is the ideal suggestion is.

You shouldn't uh like over populate your web component with lots and lots of code.

So it's the web dynpro component should properly be structured so only relevant code that you need for that particular component should be written in this.

Uh, in this part in a particular web dynpro component and suppose you.

See that some of the code can be reused.

Then you can you have an option to create a class.

And that we using that particular class methods, we can call that particular methods in multiple web Pro components.

So that's where concept of re usability comes into picture.

So how can we do that?

So if you see in this particular.

Component, there's a thing called assistance class.

So assistance class.

As the name suggests, it assists the web dynpro component in doing some of the activities, so it helps release load on the Web Pro component.

Now how it works, I'll show you that.

So what you have to do is when you create a component, you have to write the name of the assistance class.

Let's take.

The naming convention is saying that we have to use head of ICL.

Then you have to click enter and it will ask for creating the class.

And if we go to that particular class now.

So this class opens up.

So what this class is in?

This class is basically a subclass of standard Web pro class that has the.

Some existing standard declarations of the.

Which are defined by ASAP, so it has a reference to that.

So you don't have to bother about the superclass.

Whatever feature that you want to do, you can write here.

Quality code.

Say something.

It was declared as private, it went.

So now.

This was my component, so if you see a new folder gets created which has the reference to an assistance class.

So whenever you get an component and you see assistance class.

So basically there are some features that is written in this class and it might be getting called.

The component level, so now how can we call this so I'll open the component controller.

If you see there are these.

Which are already there whenever you create a web template component.

So WDS basically points to your assistance class.

This you can consider as a pointer.

WD this that could be to.

That particular.

Instance like if you are at view then WD this points to that particular view or component or window whatever it is.

And WD context points to your context.

Nodes attributes method.

So if you want to access your.

Uh, assistance class method then that you can do using WD assist.

This video.

Is this an option in the code wizard?

I think it's not there.

So if you see using that particular pointer I can navigate to my any of the assistance class methods.

So it's nothing but structuring your code.

And having separate logic at a separate method or not instead of just bombarding your particular method with all piece of code you can do.

Some of the logic add assistance class level.

Let's see this.

We can use this thing.

There's some defense.

Billy. Billy.

This will not work because I am trying to use some of the declarations which are only present at the web input level.

Maybe I can write some different logic like?

Maybe on the user I want to?

Put some or next second example of updating my log table.

So maybe I can write the logic to fetch the data here and for updating I can write a logic in assistance class to update the.

A staging table and I can call that.

So I have one example handy which I can show.

See in this assistance class.

So I have written this particular method.

To check some data provided by the component whether it's present in this particular table or not.

I have some data.

You see, they serve you.

So in this particular component I am I entering my user ID and the password and when I click on submit button validation gets performed and if validation is successful we fetch the data and display in this particular table.

And if you see the action.

Can somebody boot it?

There is a lot of disturbance.

So if I click on this submit button.

What it does this?

It calls.

The validation method that is there in my assistance class.

And returns a parameter and based on the value we can perform our further.

Like binding the table that I've done using this one and we can proceed ahead.

So let's run this application.

How's my assistance class?

No, I'm into my assistance class.

It checks for the validation.

The validation is successful.

But it returns the table with the data.

And I'm doing the binding.

So basically if I see my component, there's no such logic written in my component level.

It is at a particular class level and the same class can be used in multiple web dynpro components.

So the concept of re usability comes here.

Getting the data.

So, have you understood the purpose of assistance labs?

Any questions?

OK.

So assistance class is very much useful, so you should.

Whenever you are practicing as well try to use the assistance class because when we do code review so at that time we prefer that your code is structured and most of the reusable logic you are writing in the assistance class.

One of either.

You may also think that what is the purpose of assistance class when it's nothing but a different class.

So I can write my logic in a particular class and then maybe I can access the method of those class instead of writing a separate.

Assistant class.

So yes you can do that.

Maybe you can create static methods or.

You can write some logic there and then you can access your class here.

But assistance class basically as you can see when you create it's visible here, so at one particular.

Point only.

You can see all the related components of a web dynpro thing at one.

1.1 location itself, so that is one of the benefit.

Then you don't have to write like separate.

Chances to access each of the methods.

Like if you're creating a class and you have to access the instance method, then you have.

To create the instance of that particular thing and then that instance.

Uh, object you have to pass here and there and you have to refer that and write your logic instead you can just refer to WD as this assist and access your methods. So those are the.

Nominal benefits, but.

Structuring is.

The primary thing and availability at one location is the.

Primary benefit of assistance class.

OK now one thing is these components that we have defined are like quite simple ones.

So there is not much complexity in the code.

There might be a thing that you have a quite heavy web template component.

Hello, can you hear me?

Hello, milk?

Yeah, can you hear me?

Yeah. Now we say yes.

I guess.

I'm sorry, I didn't see that I was out.

So what was the last thing that?

So explaining why we should go for an assistance class, not creating any other.

Normal class because the instance is ready at the web control level.

OK.

OK so next thing what I was saying is suppose now we what we have designed is basically a simple web dynpro components wherein there's one view or maybe 2 views with some basic elements.

You might get a requirement that there is some complex design and.

You need to track a particular attribute like how its value got manipulated and we need to write some logic.

By identifying a piece of code which changes the value.

So at that moment what you will have?

Is some design similar to this one?

So many views, so many methods assistance class having.

Plenty of methods in that.

So it's not that simple.

So maybe you can start with a particular window, the main window or the if the.

The technical help you are getting to know the view name, then maybe in that view you can start with the init method.

Put a breakpoint there.

And then you can start on the trace so.

Let me again execute my application.

Hope the CP debugger is not running OK.

Basically once you are in debugger window.

So the whole web dynpro application design that you can see as a form of class.

So maybe 1?

Quick trick is to search for your.

Uh, that attribute name maybe?

Let me search for like.

IV under score.

So if you see if I have that attribute name then maybe first hit and trial thing I can do is I can start the.

For Windows init method, I can put a breakpoint and then I can search for the attribute name.

In this end, wherever you can see the particular variable name, maybe you can put a breakpoint and then you can proceed with your further analysis.

So here.

Most of your methods would be visible as of.

Alex's traffic.

This is the thing was like I can put a breakpoint statement which again is not advisable since you might forget to.

Remove the breakpoint and what happened during this particular thing was.

See if I I forgot to remove the breakpoint statement and.

When I executed the breakpoint state got.

OK.

So now these are some of the topics.

Any questions till now?

OK then, now let me open one document that I've prepared.

So these particular methods which are visible in.

Web denser component at view window component level, so those are basically called as hook maps.

So hook methods are basic.

As the definition says, the hook methods are basically the methods which are hooked to a particular.

Event or a particular action.

So you need to understand the flow of the hook methods and its benefit like when we can.

We should use a particular hook method and what logic should I write?

Because you may write your logic in init method.

Off of you or may you may also right.

You're logging window component level the IT would get triggered, but you must understand that each of the purpose of each of the four that I have.

We tried one document, so I'll share that as well.

But it serves the whole purpose.

It has one point.

Everything is defined.

So these are the hook methods of Web app.

So add component we have in it, then exit then application state change before navigation and post processing at controller we have these again in it.

So init and exit are at all of the components.

All of the controllers basically.

I guess the document has not loaded.

The Word document hasn't opened yet.

Yeah, I can see it now.

Thank you.

So you are able to see the.

Word document now.

OK, so as you can see at each of the controller level we have any 10 exit.

And at the view level we have like before action.

No, I had called like so.

Only you can do some action.

There's nothing that you can do.

Particularly because interacts for interaction you must have a view.

So we have the before action after the action there.

And then there's a model.

It's ideal purpose I will show with one.

It's basically, uh, OK, let's proceed and then I'll show.

Then add the component level and again on the window.

It's very much simple like window open or window.

Close only these three methods.

3 relevant methods are there at window that controller these are there and add the component controller we have like before navigation goes to say as state change so.

Let's start with play.

What happens in the post processing?

So it's those are basically required for the cleanup processes.

So basically you might have defined some references or some declarations that now.

You won't require further further processing then you can do the cleanup of that and then before navigation again.

So if we want to do something before navigation, you can do that particular thing here.

And after a particular action gets triggered and before that ultimate thing to happen you want to do something.

So you can do that particular app logic in after action and before action you have these things so the relevant thing that you can do is written in this.

You can refer this one.

And once the window opens, or.

When so window open and close so this thing is like more relevant and then suppose you are giving a pop up window.

So when you're showing declaring a window like in our code we have we had written like LR Open window open.

So if you want something to be.

Written before the window opens.

Then you should write the code in the WD WD Dio open method and after the close thing happens again for clearing or something you can write the logic and the close thing so that will get triggered at that particular moment only.

OK, now this application state change so it's basically uh relevant like your application is open and it's it is in the same state, you are not using that particular window.

For a longer time then maybe it sets to a like in web dynpro term we call it like suspending and resuming.

So at that particular moment if you want to something then you can.

Uh, make use of the state change methods, and finally comes the modify view.

So the purpose of modify view is.

Suppose you want to.

Make some changes at the view level, like some element changes at the view level.

Based on some condition or something turn idle.

Approach is to write your code in the modify view part only.

OK so as it is mentioned that this this is the only method in which dynamically modify the UI controls is permitted.

So only here whatever things you do.

Will be.

Affecting your UI view UI elements.

If you are adding somewhere else then it might not impact.

So the important methods that you would mostly use is.

The init methods.

The modified view.

And the open or close.

So these are the four or five methods that.

You would have a book called mostly.

Otherwise you might have the code in your specific actions.

Uh, like action handlers in that the code would be there.

Otherwise most of the code would be your init.

And these methods.

Otherwise you have your assistance class thing.

OK, I'll share this document with you.

Uh, we are almost done for today, but I'll show one example wherein.

We are using the modify view.

So I'm at the screen, everyone can see my simple screen, right?

OK.

So this is one of the design of web input component I have developed.

I'll first run this particular application.

So if you see this is my web dental component, so it's quite simple.

We have created a button, maybe this would be a label and this is.

My input field.

So let me go to the view.

So if you see, there's nothing at.

My view level in the layout I haven't created any new element but.

If I I was executing the application I could see these three elements.

So how was this possible?

So in this particular example.

Two things are there.

One is dynamically creating the UI elements and the other thing is.

The modify view.

So here there are some standard.

Classes with which I can create the UI elements.

So for each of the element as I told there are some classes.

So for the button there was this particular class and it has a static method to create the UI limit.

So I have used that.

Then again.

For the button add some text and those things these codes you might get.

I've even I've done the Google those things.

Otherwise you can do some demo applications.

There you would get.

So basically by referring to those code I have created these UI elements.

OK so this is one way wherein you can create dynamic UI limits now as I told.

Copy this code and write it.

At UI element level.

Sorry, init level.

OK, so now my modify view code won't get called.

As we know that the first method before the view is displayed, the first method that gets called is.

The init method.

So I am under the impression that if I write a code same code here since my view is not already declared.

Not already visible then maybe I can write a code.

To modify or maybe add UI element.

And the code I can write in the init method.

So let's see if there is some declaration error.

So yeah.

So if you see.

The difference between 2 methods.

The reference to the view is only available in the modify view method, no other.

Method would be having a parameter.

OK.

So no other method would be having.

Reference to this particular thing.

There's no parameter itself.

So that is the reason.

We cannot write the logic for modifying the view in any other method.

So this was one example where in.

We are dynamically creating a UI element at the modified value.

You can try on these examples and then maybe you can understand more so.

So again here the purpose was like we can.

Design some applications in this way as well.

So wherein we can dynamically create your element.

One maybe?

We can modify the layout of a view dynamically.

And some of the demo applications you can get.

Getting demo here.

So here you see there are some.

Demo applications for.

Any of the UI element values suggest.

Tree control some complex elements.

Something related to menu bar.

For a table, if you want to see then there's something related to demo.

So what other things you can do with the table?

So you would have an example on this one.

So if I run this one.

We're having something related to checkbox.

Table sort and filter function so we have sorting.

Suppose you have a requirement so you can do these things.

So you can refer to this demo.

You can see like how it was defined here and then you can refer the code in your application.

Similarly you just have to do start, demo, start and then maybe.

For a specific UI element you want something give some text of that particular UI element.

Then you would have some similar example to this.

And you can refer the code from that otherwise since web dynpro is like.

It has reached a saturated point, so you ideally have.

Everything available on Google for each and everything.

So you can refer the code from there as well.

Only thing you must know like what all things you can do.

How to do you have a reference for that?

Because while doing the functional FS analysis or writing the TS you have to write understand like what all things are feasible.

You must know, like what all?

Things are possible.

OK, so.

That's it for.

Any questions?

When there is one question in the message.

Box that how can we make different tabs having different views on the same window.

Yes, we can have that.

So for that.

You first.

Have to.

If you know how to create tabs, then you can.

Created tab.

Uh, I've been through component.

Take an example.

Or tab then to have separate views in the.

In that there you have view container.

If I open this one.

OK, let me.

Show this.

What was your application wherein you had done view navigation?

Uh, why?

Why APR star demo two star?

Start dimmer 2 star.

Yeah, the navigation.

OK, so.

I want to explain with the tab thing, but I would explain like how you can have multiple views in the same view.

So if you see this is the view container.

So if you see in the main.

This shows the view and this view has a view container.

And this I will have to embed the view.

So now my one view contains 2 views.

So similarly in your view you might have multiple tabs.

So in each of the tabs create separate view containers.

Then come to the window, you will be able to see those view containers and then embed the views that you want.

So to create a tab, there is a tab strip UI element.

So when you create a UI limit using tab strip, so within those tabs you can create such view containers and invade any number of views.

In a single view.

We are not showing that now because.

Controlling the tab visibility and those things it requires some.

So the layout is a bit distorted since I had directly mapped the view container instead of taking that in a separate layout or something.

So let's take a headline code.

Was there any specific code error?

Was there any?

Specific side.

Uh, yeah, take a.

End connection number.

I think that's also a no.

Just check this out Sir.

The table doesn't have much data.

OK.

So now if you see the 2nd.

Text 0017.

Yeah, yeah.

OK, this is screen now if you see in the first view itself.

I am able to.

See both the views.

And again, the visibility I could control like on the get detail only if data is found.

Instead of navigating I could have.

Displayed this particular view.

So in the design what you could have seen is.

I have the input selection screen.

So the design could be I have input selection screen and a view container.

That's it.

This once you try you will understand more.

So the view container holds multiple views.

I could have created like 3 view containers here, here and then.

Those might have multiple views in the same.

So again, that helps in structuring, like separate design could be done and there might be a requirement.

Let's say uh.

Let's take an example of sales order.

Then maybe for billing details I could have a separate view, then maybe the delivery thing I could have a separate.

View so this one of the example we that we can design using the view container.

So hope this answers.

OK, so that's it for now.

No more questions, right?

I'll share this.

So what file that I refered for the?

Hook methods with y'all.

OK.

So we'll stop for today and we'll continue for tomorrow in the last 10.

Thank you.

Thank you all for joining.

Thank you.

Hey, good morning.

Thank you, monkey.

Thank you.