yeah, it's working. Okay, so we have a lightning component which is creating account record. So, just created this lightning component over here. We can pass the apex class under the controller, okay? And we need to implement this interface to have it on the quick action

okay this from this component we can create the account record clicking on the quick action button. So whenever we will click on the quick action button component will be invoked and it will provide us a screen where we can provide the value and we can create the account record Okay, so, let me show you an account Yeah, I have created this button. So once we will click on this button, you will have this popup here we can provide the name and the value and then we will click on save and then our record will be created. Okay. So just first I'll explain you the lightning component so that it will be easy for you to understand that thing. So yeah, so in the first line, we just need to pass the controller name, which is the apex class. Epics class will go into insert our record our data, and we just implemented this interface to be available on the quick action. Okay. Here we are, we have three attributes. This is attributes for the name, type and annual revenue, these three fields we are passing, okay. So that's why we just have these three attributes over here. type is called string and here annual revenue we have a currency because it's a currency field. Like that you can pass the List map like that you can pass over here as object like that. So for that we need to create an attribute Okay, so attribute is nothing but a variable kind of thing where we can store the our variables which is type what is the type of that and that can be accessible on the our controller and then we can pass that to the epics class. So this is kind of attributes we can create okay, we can create a kind of Boolean attribute as well as like, if we want to create a flag. If it is yours that if it is true, then we can call this section like that also, we can do it. So like string, Boolean, number, integer, string and List map that kind of the type of attributes we can create in lightning component. Okay, so for the picklist options for the picklist like this is the picklist right. So how can we show on the picklist value in the lightning components so for that for that, I just use this array attribute options. Type is equal to list and default I just passed the all the picklist values like what is the label name, this is a value. This is the we need to pass in this parameter like label we need to pass the label and then the value in the string format. This is the kind of array you can see this is the you know bracket and over here we have a multiple values comma separated right so this is the one way we can do with the picklist value, but we have a different option also to take the picklist value. So this is the simple way we can do if we have a small type of values and all then we can do it like that and this is the description of this attribute so we'll attribute will define picklist value if you want to do anything you can do query from the database and set those values. So, this is nothing weather description only we can provide a description to the understand the why we attribute we are going to use this one

okay, till now

any doubt

Yeah, so in the component

we need to pass the controller epics class name, then the interface where we are going to use this one then we need to pass the attributes. So, this is the required name and type these two attributes like tag is required for that

okay. Yeah something like a class Yeah. Yeah,

yeah, this is the epics class. Because in the component we have only the no UI and whatever logic we are defining we can implement in the controller and Lauren Lauren Hello Hello Yeah, sorry. I was just disconnected. Yep. So,

controller is it? Yes, yes,

I'll explain. Yes, yes. So yes a controller it means we need to call our epics class if it is required. So now in this component we required because we are inserting record we are doing DML and DML is not possible from the lightning component in some cases we can do like as we can do the lightning data service in that we can do without any Apex controller. But now I'm just explaining you with the apex controller because this is the required thing they will go into as to question how you can pass the data from lightning component to epics class and how we will pass the variable and parameters like that. So, that's why I'm just taking this example with the class okay.

Yeah, simple thing I will

Yes, simple thing. I will explain you later. Like as like what is the lightning data system without wearing any without using any controller how we can insert the record, but that is only limited. We cannot go with total customization into it. So so yeah.

Okay. So here,

this is the one data center tag. And this is the h1 tag, where we are passing the text like custom create account. Okay, this is the CLASSES ONLY. It's not a required thing. Like, we can pass in like title top large. Isn't this horizontal like that we just passed over here. What is the font size we can pass it style. If we want to pass the custom style over here, then we can pass it over here. But it's not necessary. If you are if you can pass directly the state in h1 then also it will be fine but I'm just explain to you that this different type of classes we have I'll ping you I'll bring you that link where you can find out the inbuilt classes from the sales forces that we can use. Yeah, so and also like that you're here is also we are passing the information. So component will start from here, we have the new tech and that we are passing this SLDS class around medium and then input what type of input we have name for the name we are taking this input field so this is the for this name. Okay. Yeah. So, you can see over here this bar it means the field is required.

So how are we going to do it?

So in the lightning input tag, we have an attribute called name over here, we need to pass the API name of that name. So on the account the name is only this is the standard name that required is called to true making it true required on the layout. If we are not filling the value, it will show the error then value like whatever value we are in we are provided. We just need to take it to the sale so that we can save it. So we don't name is this is the name you can see we have created this attribute. Yeah. So in this we are storing the value whatever value we are providing to the component. Okay. Here, here you can see we don't name and then label name. Label image means only the whatever value we are showing over here. This is the name this is the label name we need to provide and maxlend This is not a required but we just provided that what is the length of that character. And second one is let me take to you how we are storing this name value added to it or else layer just well we just for the second one we just put the combo box in the combo box, we are taking the picklist value. So this is the status and for that we are showing the list of values so that's why we just put we just took the input type is a combo box okay.

Yes, yeah. So combo box also

we need to provide the status name and then value what is the default value? I think select progress. The first is there, providing the you know, this character this text, so that is something we passed over here in progress and placeholder in the placeholder we are just passing this default values select progress and values in progress is

whatever value we are taking it

and options options is this is the option we have created for this, so, that is the option it will show the list of value from this options and then on click handle change. So on Canvas gel Joe method will work whenever we are selecting any picklist values. It will it will you know store that value on to the variable. So that is something we just calling this handle change method. Okay. Third thing is a input type is equal to number that is for annual revenue. Then what is the label name? And this is the API name of this annual revenue value is enabled when you we are created this attribute in this attribute we will going to store whatever value we are providing and formatter is currency. We need to provide the data in the currency format. So that's why we just provided this one and then like value of if we are not providing number then it will show this please enter in numbers. Okay, so

we have yeah

yeah, so we need to provide the currency number types. We need to provide a we cannot provide any text. Yeah. Because yeah, yeah, because this is the providing the combo box and it will providing the list of value. So here no chance to enter any input over here, right. In the Name field in the Name field, we can write it anything because this is the name it will not restrict to like what type of name we are providing in the name also if something like so, this is not required for the name datatype I mean text data type, because text can do anything right text can be we can play we can provide the number and text in mixed right. So that's why yeah, see? Yeah. So here we have completed the input type. Okay, so here name, status and annual revenue. We have completed all three is input how we are going to add on this component or how the UI is developed. Okay, now we will we'll go with the, the how we are they were getting the data from here and passing into the component. Okay, later on we will work on the this button.

So are we looking at

Yeah, now we are taking the value. So for example, in the input name, I just provided the name value. So it will be going to store in this V dot name and in the controller

in the controller

yes, no, no, the control is not a next class controller is like a js file where we can do our business logic over here. We can like whatever method we have written over here like on chain method okay. So this method

is this one.

This method will work whenever we are going to provide the value into that picklist value right we are selecting any picklist value this method will be invoked okay. And we are going to we just created one variable. Here we are going to store that value selected options value equal to event dot get param value. So this value we will set because we are just passing this event event dot get parameter. So whenever event is happening, whatever value is changed, we will going to store into this variable okay. Okay, and after that this attributes are going to be set into this video type. It means this value we are passing into this video type okay. So the syntax is component dot set. It means we are we are storing the value into this variable. So what is the type we know type is a attribute okay. So where in this in this attributes we are passing this selected value. Okay. Yeah, so likewise we can get the data and we can store into the different attributes okay. So and in the same function, here we can, we can create a function in the function, we need to provide the name of the function then keyword function and then component event and helper we need to pass all three keyword over here. So, like for example, we are doing component dot set. So, what is the component component is nothing weather but a keyword of that. So, based on like whatever action we are performing, it will be stored into the component dot set. So, we need to write that when an event we are using this event or parameter right. So, that's why we're just event over here. So, these three three component is required. We need to add helper whenever we are calling any method from this helper. Then we need to have the helper keyword in this function

okay yeah, these are the two

required parameters of the function

okay. So now in the same way that we are

getting all the inputs, whatever we are provided on the component level here so variable type I will create a variable type then component get. So, for example, we are setting value over here components are saved. Then if we want to get the value there we need to use this component dot get. Okay, so now we are taking the value we don't name we don't name is something

this one so we are taking

value we don't name type and annual revenue and then option okay. For three parameters we are taking over here so name we are taking this name then we are created type and then we are getting component dot get type and annual revenue we are taking it component dot get dot annual revenue. Okay, here we are checking if the data is not null or what if name not if name is equal to undefined, or type is equal. To undefined or annual revenue is undefined. Then undefined if something if it is null or if we are not providing any value or like that, if it is not a correct format, then we need to check like that undefined

Yeah, yeah. So no, no, only the first one we just provided the required equal to true for the second you can see there is no Starmark over here and for third one is also not right. So just explaining you for first one, like we can pass this request to true if we if we cannot prove some in some cases we cannot provide the Starmark so that for that we need to handle it. Custom way. Okay. Yeah. So for so for that I'm just checking over here undefined and then if it is undefined that we are passing the one error message like show toast message. So, helper show toast is a method which is created in helper class, in helper file or in this we are passing this two string, like three string when is it and please fill up all the information and error what is the type that is type is error of this toast message. So let me show you this method first. Okay. Show test function. Here we are passing I'm passing the title message and error parameter. Let toast parameter so in the last parameter, we could pass this title, message and type. So title is equal to title message message and the type is equal to error. And then here again created toast and this is the standard, you know, functionality provided by Salesforce where we can pass this e dot force dot show toast and then we need to pass the set Parameter Set param it means we are passing the set moderator and what is this post param is here we are passing the parameter likewise, inlet those parameter with the parameters we are passing title message and error. So here we are setting it into this this like get force post show toast event. And then we are firing over here this event. Toast event on fire.

Okay. Yeah. So

so we are taking three parameters over here. That's why we provided over here, three string, like first is oops, then then this is the information then type is errors okay we able to see Yeah, yeah, let me I'm just passing the test over here and then we want to see that oops, please fill up all the information and error it means it will show the red label red label a warning message. Okay. Yeah, is that clear? I mean yeah, I mean, let me explain you again. Once we will click on Save button. Save it means this is the button we have created over here. Lightning button and variant is equal to brand brand it means we are select we are taking brand so it will you know the color will be blue. And that label name is saved and title is also save and honestly we are we are have one method we created in controllers save so yeah calling like C dot save. Okay, yeah, same way we have created cancel button. Now the variant is neutral label title is cancel and on the C dot cancel. So we are now in the same water right same function over here we just take it all the input data getting component dot get right and then we are checking undefined. And then if it is undefined then we are showing this toast message error message and then in else part we are actually passing the variable all the attributes to the apex controller. So now we are calling this variable we have created one a variable called action and then component will load get c dot save account so what is the C dot save account? That is something we have created one epics class Yeah, so you can see save account. This is our epics epics method okay. So for that, whatever method should be available on the component level we need to help this keyword at edit or are enabled

okay, if you are not providing this

keyword then this method will be not accessible on the component level it will show you the errors. Okay, so here we just passing the string name, string account type and decimal revenue. Okay, because we are we are having three values from the input, right? Name, account type and revenue. So these two is a string datatype and this one is a decimal and here we are doing the insert statement. So, we just created account is equal to new account and then ACC dot name is equal to this name because we have a data into this variable, this attributes right SSA type is equal to account type this one and then SSA revenue annual revenue is equal to this revenue. Now we have all the data in our ACC attributes right and now now we are just putting in a statement over here insert ACC because now on the form side we are only inserting only one record at a time right. So that's why we simply just inserted into this. We are not taking any list and we are returning the list of returning the ACC so that it will be like now we are not using this return data but yeah just provided because it will return the account record.

So till now

any any problem yeah, this is easy something right? This part okay yeah, so I'm just explaining you so this is the method we have created. We just need to copy this save account method and in DTS file. You can see we have this one this one C dot how we can take it the question will be in India viewer will ask you how you call your epics method into the into the lightning component. So you need to be answered like we can create directly component or get c dot method name.

Okay, C dot method name.

Yeah, yeah. And we need to provide the class name in the Controller component level and then we and then method should be already enabled. Then we can write this statement component dot get c dot account name, like C dot method name. Okay. Here we just call this method but we need to pass the parameters right because this method is taking three parameters, right? And we already have that variable which is created name type in annual revenue. So all these three we need to pass with the this action. Actually it is the variable dot set params set params. It's like setting the parameters right and over here in the bracket, we need to pass the name equal to this name, account type school to this type and revenue school to this revenue type. Okay, yeah. So here we just pass over three parameters. We're passing the value over here and then then it will call to that Epic's class over here and it will insert the record and it will return the result. So how we are taking the result. So action dot set callback, and this function responses points is what response we are getting is it inserted or not? Okay, here we are. We are handling the response from this set callback method. Okay, this is the standard things we need to write every time

action dot set callback.

Okay, action is a variable we have created and dot set callback and this function and response and variable we have created one more state and responded get state what is the state of this response it is success or not. So here we are checking if State equal to equal to success. Then

we need to pass the parameter.

Then we need to show this toast message like helper dot show toast success record is inserted successfully and live kind of what is the type of that label that is success it means it will show in the green or blue right and then we are firing this event over here. So it will close the that quick action it forced dot close skip the action dot fire this is the event standard event. Okay. dollar a dot get e dot force dot close click action dot fire so here yeah, in this if statement in this part will execute and then into action into action is not required now, but I just put it down because it means whenever we have a multiple, you know, method we are calling in one one method. Okay? Then we need to write like into action. So all the our action will be make a make interview and they will execute one by one. Okay. So this is the this is the our method is completed over here. And we have a Cancel Cancel is only in the cancel method we are only doing closing this quick action like that pop up only. So this event will fire a door for the closet protection, fire and it will be closed admin popup right so in this stage

to discuss is a lightning component.

Yes, this is the lightning component

declaring all the

information yeah CMP file will show you the all the UI part. Yeah, whatever on the UI part you are showing. For example, I am going to update this one okay, custom create account.

And then just saving whatever changes we just want to

like whatever UI part we want to build that will be available in the CSV file. Here you can see the text is updated.

And based on this text

or a component that pop up size is also increased. Right. Yeah, this is the responsive because we are using the SLDS and everything. Yeah, so yeah, right. Information. Yeah. So we just took the lightning input input field, we are we are taking the input from the UI part. That's why we just use this tag. Lightning input for the name status and a number. Okay. And then this deal is closed over here and then we are going for that button. So we just took the lightning button and these are the attributes we pass that what type of button we are using label title and on click LIKE WHAT WE WILL happen on clicking on the Save button and like that we are handling this Yeah, so CMP file is CMP file is like there is no issue on this. You can ask me anything on the CMP file.

yeah yeah so I'm just putting the this one so for the for the epics level debug you can find it over here in the debug log right but for the for the J's file console look where you will find that is something I will show on the in the insect console here you can check if we are getting the value in over here or not okay,

some let me check up I have saved it I'm going to make this and there's a facing it showing the alert message no this

is the lightning This is the lightning component lightning an aura component is the same okay once I will click on Save button you will see over here the console

is not printing okay I need to create a new one not to save this setting this thing change to whatever console is not working okay, let me check yeah I think the cancer just the epics class is not required to be refreshed multiple time but lightning component may need to be refreshed to three times

yeah

I just intertext Yep, you can see we have a console name, type and enable data. So in the name we have created, you know provided test value and then the customer directory is a picklist value and annual revenue. We have selected this one in C. So here we can see over the value and in the debug log. We can check the system dot debug here. What is the name account when you and this is something coming from the trigger?

Yeah. So this is the controller. Yes

helper which required sometimes whenever we want to call our method multiple time because you can see the show test method we have called multiple times. Here we just call for providing the error. Right? And here we just show the description messages. This is the custom method we have created. I mean if that is not required if this is the best practices, if we can follow to we had all the all the errors and you know this is easy for the user to understand what is there. If we are going to clicks click on a sale button and it is not showing any message like it is record is getting saved or not or what right so that's why this is for the validation

or no, this is the customer we can we can update it. You can provide it so anything yeah you can do it anything. So this is the this is the standard you know, event we need to call Otherwise, otherwise any variable name we can change it this late you can change it. But the parameter will not get changed the parameter is titled message and type. Okay. And yeah, so that's why we are calling this test method to time. So that's why we just write down in the helper method. We are we are calling it for the success also and we are calling it from the for the error also right. So like that we can create this is a simple lightning component where we are going to just creating a lightning creating record and if we want to like how that come from the component like record should be created and record should be you know is updated and record. We have the data table in that we can show all the records from that related account. Like all the contact of that related account. So in the table how we can show the records and we have an inline editing. Once we will click on the double click on the table that inline editing should be worked and we can update multiple records and we can click on Save then all the records we have updated that should be saved to the database. Likewise there are different different component we can build it

yeah so this is Salesforce user defined this one yeah, this is also custom This is also custom everything every method is a custom but the you know syntax we need to be identified that should be function and component then even in here we I'm not calling any helper methods so I'm just not added the helper, comma helper. Here I'm just using component and event suggestion but you can by default, you can add or the component or helper components event helper

referring to

this is this is something when we are changing the value into that you know picklist value right? We are taking the position. Yeah, this is this is for the picklist value. Whatever when you have we have selected that should be come from the event dot get parameter value because how we will get to know like we have selected this one or we have selected this one right. So, like I'm going to change now I have selected prospected mmm now I'm changing this one. So how will this is the event on click event this is the standard event and clicking on this it will take the value and it will store into this this parameter so event dot get param dot value. That is the

taking the actual value whatever we have selected Yeah, so number 60s Yeah

we are Yeah we are. Yeah, yeah here we are saving our records. So that that in here we are calling our epics class method as well to passing the parameter. Let me explain you one more time. Here we need to define the same method the name should be anything you can call it save account, save account record or something then later on function event and component and helper it's mandatory. And then we are doing we are taking the value from that. Whatever we have selected, we have provided on the UI part okay? And then this is the console log if we want to print our values. Okay, and then we this is also not mandatory, but for the for the example purpose. I'm showing you how you can check if it is not null. All right, like that. Yeah. So, here, we are just taking the undefined and showing this last message for the user. And in the else part if it is not undefined and the success then we are calling our epics class method, epics class method. If is class method should be already enabled and that should be when it is taking the parameters so that we need to provide the parameter over here set parameters okay, and it will be called component dot get what is the component get and component set? Yeah, like when we will write component get and when we write component dot set

Yeah, in yeah in any any note from a big splash in any like here we are. Here we are. Yeah, here we are getting it. Component don't get we don't value we don't name it is from component file CMP file. Here we are using component get and we are calling our epics class method. Because we are taking the value from we are passing the value over there. Right. I mean we are getting the value. Yeah, yeah. And we are a components dot set. We'll be setting the value to show on the component level. So, yeah, and here actually looks at params we need to pass the three parameters because in these action, the Save account method we have three parameters. So that's why we are passing over here three parameters right. Yeah. And yeah, the left side part is from the the effects class variables and in right side part is coming from the our component level name type in annual revenue. Okay, so we are setting the effects with the company. No, now we have a value into the name into the name right and from the component from the CMP and this value, we are passing into the epics class variable

yeah

yeah, yeah. So every class will be going to be run in the backend. And likewise, we have passing the parameter was the parameter pass and our method is run successfully then we need to store the response. So action dot set callback method, we need to call over here and then we are taking the response and then we need to create one variable to check the what is the state of that response, if it is success or fail, and if it is state is equal to equal to success then we can do another you know any action if we want to call any another method or or whatever value we are getting we need to store into the another list then we can create one more list over here. Like or attribute name is equal to type and name equal to anything and type equal to list. If we have a list of records, then if we have only one record then also we can create a list it is not you know, have any issue and we can store we can we can do it component dot set and that the response we can say that responding to it and the cancel button is only for the closing that pop

up yeah, it will close that gap Yeah, so yeah, so now this is the basic you might get. Have some.

If someone will ask you the question interview question then you can be you know, able to answer that like yeah, they will ask you on the CMP level, they will ask you the question like, What type of interface we need to implement for that different different you know, to show our lightning component the different different places that is one question and what type what data type we can use when while defining any attribute. So that should be string, Boolean list number currency like that. Okay, and how we we call our controller into the CMP then you can get your like controller equal to our epics class name. And how epics class method we can call it on the same controller js file, then you will need to save that method should be aura enabled and we can call it directly component dot get c dot method name and, if it is taking any parameters, then we need to pass the parameter using set params. Okay. So, in this example, we have to fetch data. So, that's the problem we have actually no, oh here, we are not fetching we are only saving the records in the controller. Yeah, we are we are only doing DML we are not throwing the list of account to show on the component level, right. This is the one way we are just taking input from the component and we are passing it to the apex class and upper class is only storing the data into the database. So, this is the one way we are doing now, we can create one more method to show the list of account. So, here how we how we will do we can create a one method here we can throw the list of accounts and then at least we need to show on the component level so I'll explain you this our next session. So, how we can iterate list of data and how we can show on the table? multiple records Okay, so in the component level we need to check like how iteration will work and like we have a list of records in the come in this EMP we are we are getting the response and response is a list of records and how we can iterate that list and we can show on the component level

the next class we will go into Yes. So it's too many things

Yeah, you just Yeah. You know aware about the flow of components. What is the required thing we need to do? Right

yes

yeah maybe next time yeah. Yeah, just try to create any small custom component lightening components so that you will get to know like what type of issue you will face or what type of syntax error you will be have or like that so just try to be clear that when I'll show you this well as well. Okay, just try to create a new lead record insert from this component, and if you are not going to, you know, call that component from the button directly, you can add that component over here, clicking on edit that page and edit page and then you can directly add that component over here. So that component will look like only this kind of screen. Yeah yeah, but once you will take on that like oh, it should be you know, when the record is like clicking on this we have like pop up and then once the record is saved, then we are showing that message and then closing the pop up and then we get to know yes the account is created like that. So yeah, the same way you can also try to build like that. So, you know will be easy for you to understand that flow. You need to be have a more practice on it. So like so that will be you know, kind of hands on for you you can also stop the recording Yep.