# Create a Todo App in SAP Build Part 4 – Edit and Delete Todos

This case assumes you completed Part 2. In this case, you will create a page that allows the user to edit and delete Todos.

Create a new page called Edit Todo and create a form like you did for the New Todo page.

Graphical user interface, text, application

Description automatically generated

We will need two buttons so drag a Row control below the form. The Row control allows you to divide the interface into columns. There are two columns by default.

A picture containing diagram

Description automatically generated

Drag two buttons into the two columns and configure them as shown below. Use the Style tab in the Properties to change the style of the Delete button.

Graphical user interface, text, application

Description automatically generated

Click Data.

Graphical user interface, application, Teams

Description automatically generated

When the user taps a todo on the list, this page will open and the todo will be retrieved from the backend. If the user edits the todo and saves it, the todo will be saved and if the user taps the delete button, the todo will be deleted.

You need to activate the GetTodo, UpdateTodo and DeleteTodo functions.

Graphical user interface, text, application

Description automatically generated

We need to pass the id of the tapped todo from the list page to the edit page. We use a Page parameter to do that. Click Variables.



Click Page Parameters

Graphical user interface, application, Teams

Description automatically generated

Then click ADD PARAMETER.

Logo

Description automatically generated with medium confidence

Select the parameter.

Graphical user interface, text, application, email

Description automatically generated

Configure the properties as shown.

Graphical user interface, text, application, email

Description automatically generated

Now we need to create a Data variable to contain the data retrieved from the backend system. Click DATA VARIABLES.

Graphical user interface, application

Description automatically generated

Click ADD DATA VARIABLE and configure the data variable as shown.

Graphical user interface, text, application

Description automatically generated

Click the X icon below the ID to bind the Page parameter to the parameter the GetTodo function requires.

Graphical user interface, text

Description automatically generated

Select Data and variables.

Text

Description automatically generated with medium confidence

Then Page parameters.



Select the Page parameter.

Graphical user interface, text, application

Description automatically generated

Open the logic pane by clicking the black bar at the bottom of the screen.

Graphical user interface, text

Description automatically generated

When the page loads, the Get record component will retrieve the Todo and set the data into the Data variable.

The Delay logic component will cause the Todo to be refreshed every 5 seconds. That will cause a problem unless the user is really fast. You can either increase the Time to wait or you can delete the Delay.



Return to the View and bind the Input fields as you did in the Create case.

Graphical user interface, text, application, email

Description automatically generated

Select the Save button and open the logic pane.

A picture containing application

Description automatically generated

This logic is similar to the create logic. The difference is that you use an Update record component instead of Create record component.

Graphical user interface

Description automatically generated

Bind the ID (click the ABC icon) to the id of the Data variable.

Graphical user interface

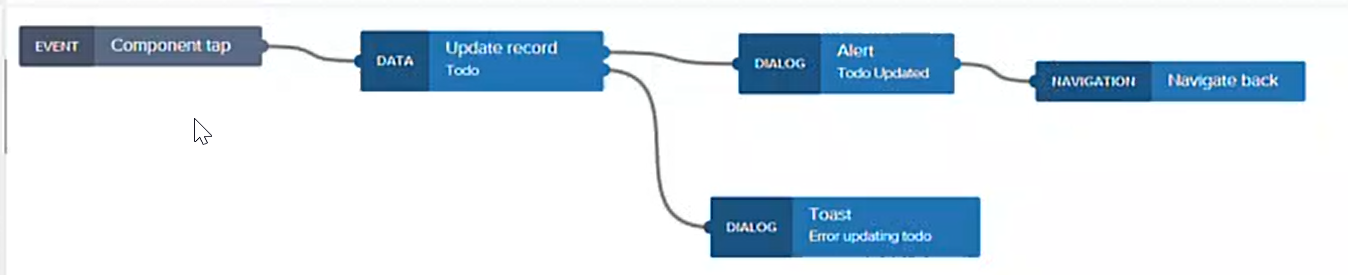
Description automatically generated with medium confidence

Bind the Todo (click the icon with the two curly brackets beside Custom object) to the Data variable.

Graphical user interface, application, Teams

Description automatically generated

Finish the configuration of the logic in the same way you did for the Create todo.



Configure the logic for the Delete button as well.

Graphical user interface, application, Teams

Description automatically generated

Next, we have to configure the list on the My Todos page so when the user taps and item, it navigates to the Edit Todo page.

Open the My Todos page, select the list and open the logic pane.

Graphical user interface, application

Description automatically generated

Drag an Open page component onto the pane.

Graphical user interface, application

Description automatically generated

Configure it to open the Edit Todo page.

Graphical user interface, application

Description automatically generated

Click the X icon below the TodoID parameter.

A picture containing text

Description automatically generated

We need to bind this to the id of the todo that was tapped. Select Formula.

Graphical user interface, text, application, whiteboard

Description automatically generated

Click Create formula.

Graphical user interface, application, Teams

Description automatically generated

Select Selected component properties.

Text

Description automatically generated with medium confidence

Select self.tappedItem.id then click Enter +.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Now Launch the app. If you tap on a todo in the List:

Graphical user interface, application

Description automatically generated

The Edit Todo page opens with the Todo’s data:

Graphical user interface, application

Description automatically generated

Edit the data and save.

Graphical user interface, application

Description automatically generated

Et voila!

A picture containing graphical user interface

Description automatically generated

Test the delete as well.