

# Exercise 5: Create a Wizard

Exercise to Accompany  
Create a Wizard and Save Your Data

The course Interfaces 102: Building Dynamic Interfaces contains 4 exercises. These exercises build upon each other. Complete exercises in order and keep the app and all objects until you are done with the course.

- 1 Set up Your Workspace
- 2 Store Your Data
- 3 Capture User Selections
- 4 Create an Editable Grid
- 5 Create a Wizard**

<b>Exercise 5: Create a Wizard</b>	<b>2</b>
Add the Milestone (Bar) Pattern	3
Populate the Wizard Steps	4
Save Your Data to Rule Inputs	7

### **Notice of Rights**

This document was created by Appian Corporation, 7950 Jones Branch Dr, Tysons, Virginia 22102. Copyright 2025 by Appian Corporation. All rights reserved. Information in this document is subject to change. No part of this document may be reproduced or transmitted in any form by any means, without prior written permission of Appian Corporation. For more information on obtaining permission for reprints or excerpts, contact Appian Training at [academyonline@appian.com](mailto:academyonline@appian.com).

## Exercise 5: Create a Wizard


In the previous exercises, you built an interface that allows mechanics to select a vehicle maintenance request, view its details, and create a list of parts needed to complete the maintenance. In this exercise we will create a wizard that will enhance the usability of the interface by breaking the process into three steps: Select a Vehicle, Add Parts, and Review & Submit.

At the end of this exercise, your interface should look something like this:


Select Vehicle

Add Parts


Review and Submit



**2005 Buick Rainier**  
The transmission fluid needs to be changed and the filter replaced to ensure proper functioning of the transmission.



**2008 Chevrolet Express**  
The brake pads are worn and need to be replaced.



**2000 Plymouth Breeze**  
The vehicle is due for an oil change and tire rotation, said the mechanic.

CANCEL

NEXT

Select Vehicle

Add Parts

Review and Submit

Part	Unit Cost	Quantity	Cost	
Rear Door Panel	175	1	175	×
Rear Door Hinge	45	2	90	×

[Add Item](#)

BACK

CANCEL


NEXT

Select Vehicle

Add Parts

Review and Submit

**Maintenance Information**



**2005 Buick Rainier**  
The transmission fluid needs to be changed and the filter replaced to ensure proper functioning of the transmission.

**Parts**

Part	Unit Cost	Quantity	Cost
Rear Door Panel	175	1	175
Rear Door Hinge	45	2	90

BACK

CANCEL

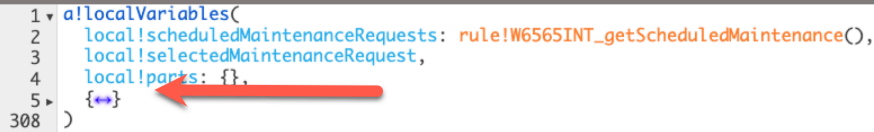
Estimate Total: 265

SUBMIT

## Create a Wizard

In this step, you will use the SAIL expression below to create a wizard for your interface consisting of three steps.

1. Copy the text below.
2. Paste it into your interface expression just above the `a!forEach()` function you added in a previous exercise.



```
1 a!localVariables(
2   local!scheduledMaintenanceRequests: rule!W6565INT_getScheduledMaintenance(),
3   local!selectedMaintenanceRequest,
4   local!parts: {},
5   { }
308 )
```

```
a!localVariables(
  local!milestoneBarSteps: {
    /*Replace the text below with the step titles.*/
    "Step One Title",
    "Step Two Title",
    "Step Three Title"
  },
  local!currentMilestoneBarStep: 1,
  {
    a!milestoneField(
      steps: local!milestoneBarSteps,
      active: local!currentMilestoneBarStep,
    ),
    choose(
      local!currentMilestoneBarStep,
      {
        /*Replace the a!richTextDisplayField() below
        with the contents of the first step.*/
        a!richTextDisplayField(
          value: "The contents of the first step go here.",
          align: "CENTER"
        )
      },
      {
        /*Replace the a!richTextDisplayField() below
        with the contents of the second step.*/
        a!richTextDisplayField(
          value: "The contents of the second step go here.",
          align: "CENTER"
        )
      },
      {
        /*Replace the a!richTextDisplayField() below
        with the contents of the third step.*/
        a!richTextDisplayField(
          value: "The contents of the third step go here.",
```

```

        align: "CENTER"
    )
}
),
a!columnLayout(
    columns: {
        a!columnLayout(
            contents: {
                a!buttonArrayLayout(
                    buttons: {
                        a!buttonWidget(
                            label: "Back",
                            value: local!currentMilestoneBarStep - 1,
                            saveInto: local!currentMilestoneBarStep,
                            style: "OUTLINE",
                            showWhen: local!currentMilestoneBarStep > 1
                        ),
                        a!buttonWidget(
                            label: "Cancel",
                            style: if(
                                local!currentMilestoneBarStep = 1,
                                "OUTLINE",
                                "LINK"
                            )
                        )
                    },
                    align: "START"
                )
            }
        ),
        a!columnLayout(
            contents: {
                a!buttonArrayLayout(
                    buttons: a!buttonWidget(
                        label: if(
                            local!currentMilestoneBarStep=length(local!milestoneBarSteps),
                            "Submit",
                            "Next"
                        )
                    ),
                    saveInto: if(
                        local!currentMilestoneBarStep=length(local!milestoneBarSteps),
                        {
                            /*Use a!save() to save values to rule inputs
                                when form is submitted.*/
                        },
                        a!save(
                            local!currentMilestoneBarStep,
                            local!currentMilestoneBarStep + 1
                        )
                    )
                )
            }
        )
    }
)

```

```

    )
    ),
    style: "SOLID"
  )
}
)
}
)
}
)
}
)
}
)

```

The image shows a wizard interface with three steps: Step One Title, Step Two Title, and Step Three Title. Step One is the active step, indicated by a blue line and a blue arrow. Below Step One Title, there is a 'CANCEL' button and the text 'The contents of the first step go here.' Below Step Two Title, there is a 'NEXT' button.

## Populate the Wizard Steps

In this step, you will move the components that you have previously created into the correct steps of the wizard, and create content for the third step: Review and Submit

1. Locate the `choose()` function and delete the contents of the first step. Replace these with the following:
  - a. The **`alforEach()`** function you created earlier that generates the maintenance request cards.
  - b. The **`sectionLayout()`** component that displays the maintenance summary view.
2. Delete the contents of the second step and replace with the **`algridLayout()`** component for adding parts that you created in the previous exercise.
3. Replace the contents of the third step with:
  - a. An **`alsectionLayout`** component labeled “Maintenance Information” that displays details from the **`local!selectedMaintenanceRequest`** variable.  
In the sample expression provided at the end of this step, the maintenance information is presented in a card similar to those created in Exercise 2. The existing **`alcardLayout`** was copied with references to **`fv!item`** replaced with references to **`local!selectedMaintenanceRequest`**.
  - b. An **`alsectionLayout`** component labeled “Parts” that contains a read-only grid displaying the information contained in the **`local!parts`** variable.
    - i. Set the *data* parameter for the grid to **`local!parts`**.
    - ii. Set the *columns* parameter to a list of four **`alGridColumn`** functions labeled “Part”, “Unit Cost”, “Quantity”, and “Cost”.
    - iii. Set the values for the columns by indexing into **`fv!row`** using the square bracket notation for record type fields.
  - c. An **`alrichTextDisplayField`** component that displays the total cost of all parts stored in the **`local!parts`** variable.  
In the sample expression provided at the end of this step, the rich text field

contains an **if()** statement to handle instances where the **local!parts** variable is empty.

4. Replace the placeholder values for the step labels in the local variable **local!milestoneBarSteps** with the following:

```
{"Select Vehicle", "Add Parts", "Review and Submit"}
```

The third step of your wizard should now look something like this:

Part	Unit Cost	Quantity	Cost
Rear Door Panel	175	1	175
Rear Door Hinge	45	2	90

Estimate Total: 265

Below are examples of what the contents of the **choose()** function should look like at this stage.

The first two steps are populated with the **a!forEach()**, **a!sectionLayout()**, and **a!gridLayout()** created in previous exercises:

```
175 ▾ | | | | choose(  
176 | | | | local!currentMilestoneBarStep,  
177 | | | | {  
178 | | | |   a!forEach(↔),  
238 | | | |   a!sectionLayout(↔)  
243 | | | | },  
244 | | | | {  
245 | | | |   a!gridLayout(↔)  
310 | | | | },  
    | | | | }
```

The third step is populated with two **a!sectionLayout()** components and an **a!richTextField()**. The first section should look like this:

```

311 ▾ a!sectionLayout(
312     label: "Maintenance Information",
313     contents: a!cardLayout(
314         contents: {
315             a!sideBySideLayout(
316                 items: {
317                     a!sideBySideItem(
318                         item: a!imageField(
319                             label: "Image",
320                             labelPosition: "COLLAPSED",
321                             images: {
322                                 a!webImage(
323                                     source: local!selectedMaintenanceRequest[W0000AA Maintenance.aaVehicle.imageUrl]
324                                 )
325                             },
326                             size: "MEDIUM",
327                             isThumbnail: false,
328                             style: "STANDARD"
329                         ),
330                         width: "MINIMIZE"
331                     ),
332                     a!sideBySideItem(
333                         item: a!richTextDisplayField(
334                             labelPosition: "COLLAPSED",
335                             value: {
336                                 a!richTextHeader(
337                                     text:
338                                         local!selectedMaintenanceRequest[W0000AA Maintenance.aaVehicle.year]&" "&
339                                         local!selectedMaintenanceRequest[W0000AA Maintenance.aaVehicle.make]&" "&
340                                         local!selectedMaintenanceRequest[W0000AA Maintenance.aaVehicle.model],
341                                     size: "MEDIUM"
342                                 ),
343                                 a!richTextItem(
344                                     text: {local!selectedMaintenanceRequest[W0000AA Maintenance.issue]}
345                                 )
346                             }
347                         )
348                     )
349                 }
350             )
351         },
352         height: "AUTO",
353         style: "TRANSPARENT",
354         shape: "ROUNDED",
355         marginBelow: "STANDARD"
356     ),
357 )

```



The second section should look like this:

```
358 ▾ a!sectionLayout(  
359   label: "Parts",  
360   contents: {  
361     a!gridField(  
362       data: local!parts,  
363       columns: {  
364         a!gridColumn(  
365           label: "Part",  
366           value: fv!row[ W0000INT Part.name ],  
367         ),  
368         a!gridColumn(  
369           label: "Unit Cost",  
370           value: fv!row[ W0000INT Part.unitCost ],  
371         ),  
372         a!gridColumn(  
373           label: "Quantity",  
374           value: fv!row[ W0000INT Part.qty ],  
375         ),  
376         a!gridColumn(  
377           label: "Cost",  
378           value: fv!row[ W0000INT Part.unitCost ]*fv!row[ W0000INT Part.qty ],  
379         )  
380       }  
381     ),  
382     a!richTextDisplayField(  
383       align: "RIGHT",  
384       value: a!richTextItem(  
385         style: "STRONG",  
386         text: "Estimate Total: "&  
387         a!currency(  
388           isoCode: "USD",  
389           value: if(  
390             condition: a!isNotNullOrEmpty(local!parts),  
391             valueIfTrue: sum(local!parts[ W0000INT Part.qty ]*local!parts[ W0000INT Part.unitCost ]),  
392             valueIfFalse: 0  
393           )  
394         )  
395       )  
396     )  
397   }  
398 )
```

## Save Your Data to Rule Inputs

In this step, you will configure the Submit button on the final step of the wizard to save the data from your local variables into rule inputs so that it can be passed to a process model.

1. Create a rule input.
  - a. Name the rule input "parts".
  - b. Set the type to the Part record type created in Exercise 1.

- c. Check the **Array** box since there may be multiple Part records generated with each maintenance estimate.

**New Rule Input**

**Name \***  
parts

**Description ?**  
Array of parts added to maintenance estimate  
44/500

**Type \***  
W0000INT Part X ☒ Array (multiple values)

CANCEL CREATE CREATE AND ADD ANOTHER

2. Locate the **Next/Submit** button in the interface expression. It should be near the bottom. The *saveInto* parameter of this button is set to an **if()** function. The first argument of this function determines whether the user is on the last step of the wizard. If so, the next argument will execute. Currently this argument contains the comment:  

```
/*Use a!save() to save values to rule inputs when form is submitted.*/
```
3. In the second argument of the **if()** function, create an **a!forEach** loop that will set the *maintenanceId* field for each item in **local!parts** to the *requestId* of **local!selectedMaintenanceRequest**. This ensures the Part records, when created, are associated with the correct Maintenance record.
  - a. Set the *items* parameter to **local!parts**.
  - b. Set the *expression* parameter to:
 

```
a!save(  
  target: fv!item['W#INT Part.maintenanceId'],  
  value: local!selectedMaintenanceRequest[W#AA Maintenance.id']  
)
```
4. Add another **a!save** function after the **a!forEach()** loop to save the values from **local!parts** to the rule input **ri!parts**.

5. Test that clicking **Submit** updates the rule input and that the *maintenanceId* field is set.

▼ RULE INPUTS <span>+</span>	
Name	Value
[-] parts (2 items)	[[INT Part name=Rear door panel, unitCo...
[-] [1]	[INT Part name=Rear door panel, unitCos...
● name	Rear door panel
● unitCost	175
● qty	1
● maintenanceId	118
[-] [2]	[INT Part name=Rear door hinge, unitCos...
● name	Rear door hinge
● unitCost	45
● qty	2
● maintenanceId	118

Here is an example of what your interface expression should look like at this stage:

```
1 ▾ a!localVariables(  
2   local!scheduledMaintenanceRequests: rule!W6565INT_getScheduledMaintenance(),  
3   local!selectedMaintenanceRequest,  
4   local!parts: {},  
5 ▾ {  
6 ▾   a!localVariables(  
7 ▾     local!milestoneBarSteps: {↔},  
12 ▾     local!currentMilestoneBarStep: 1,  
13 ▾     {  
14 ▾       a!milestoneField(↔),  
18 ▾       choose(↔),  
243 ▾       a!columnsLayout(  
244 ▾         columns: {  
245 ▾           a!columnLayout(↔),  
269 ▾           a!columnLayout(  
270 ▾             contents: {  
271 ▾               a!buttonArrayLayout(  
272 ▾                 buttons: a!buttonWidget(  
273 ▾                   label: if(  
274 ▾                     local!currentMilestoneBarStep=length(local!milestoneBarSteps),  
275 ▾                     "Submit",  
276 ▾                     "Next"  
277 ▾                   ),  
278 ▾                   saveInto: if(  
279 ▾                     local!currentMilestoneBarStep=length(local!milestoneBarSteps),  
280 ▾                     {  
281 ▾                       a!forEach(  
282 ▾                         items: local!parts,  
283 ▾                         expression: a!save(  
284 ▾                           fv!item[ W5656INT Part.maintenanceId ],  
285 ▾                           local!selectedMaintenanceRequest[ W5656AA01 Maintenance.id ]  
286 ▾                         ),  
287 ▾                       ),  
288 ▾                       a!save(  
289 ▾                         ri!parts,  
290 ▾                         local!parts  
291 ▾                       ),  
292 ▾                     },  
293 ▾                     a!save(  
294 ▾                       local!currentMilestoneBarStep,  
295 ▾                       local!currentMilestoneBarStep + 1  
296 ▾                     ),  
297 ▾                   ),  
298 ▾                   style: "SOLID"  
299 ▾                 }  
300 ▾               }  
301 ▾             }  
302 ▾           }  
303 ▾         }  
304 ▾       }  
305 ▾     }  
306 ▾   }  
307 ▾ }  
308 )
```