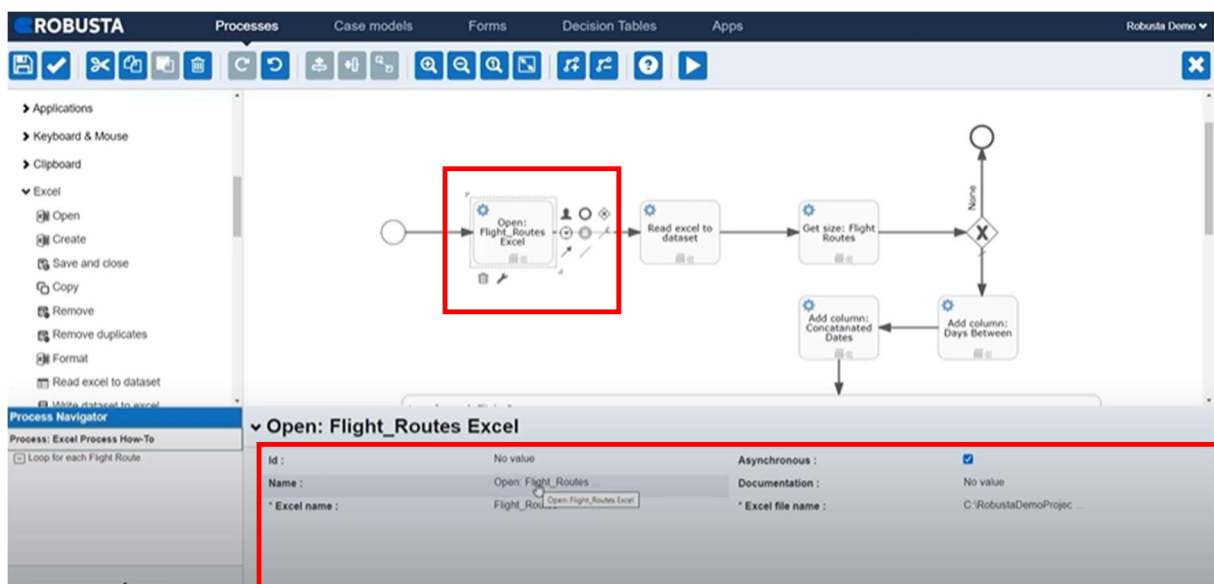


# Excel Automation using with Robusta RPA

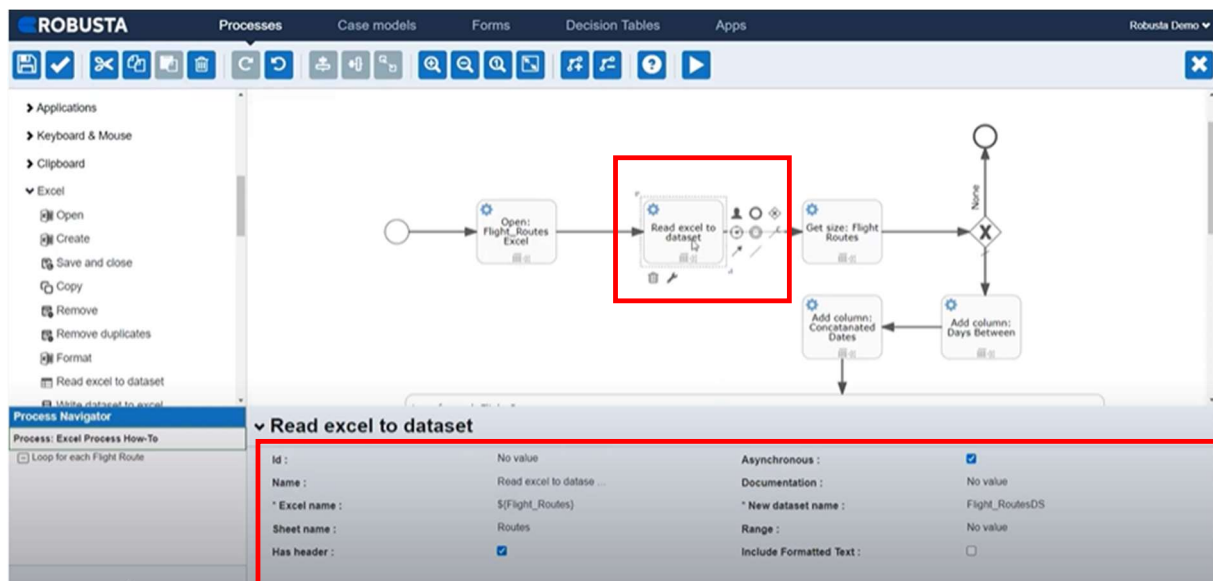
- In this tutorial, we will show you how to design a sample process that includes Excel operations with Robusta RPA, step by step.
- Our aim in this tutorial is open an Excel file and check whether there is any flight routes data in it. If there is, we will add two columns, one for calculating the number of days between departure and return dates, and one for combining these dates. Then, we will continue with a repeating sub-process since we want to perform operations for each row in the Excel file. In this sub-process, we will set the values for the newly added columns.



Excel > Open Activity	
Name	OpenFlight_Routes_Excel
*Excel Name	Flight_Routes
*Excel File name	<a href="#">Current directory where the file is located</a>

**P.S:** In this field special character constraints is important, only letters and numbers are recommended

- We started our process by using the Open activity under the Excel section to open the Excel file that contains the flight routes data. Just drag and drop the activities you want to use into the design area then enter the relevant parameters.



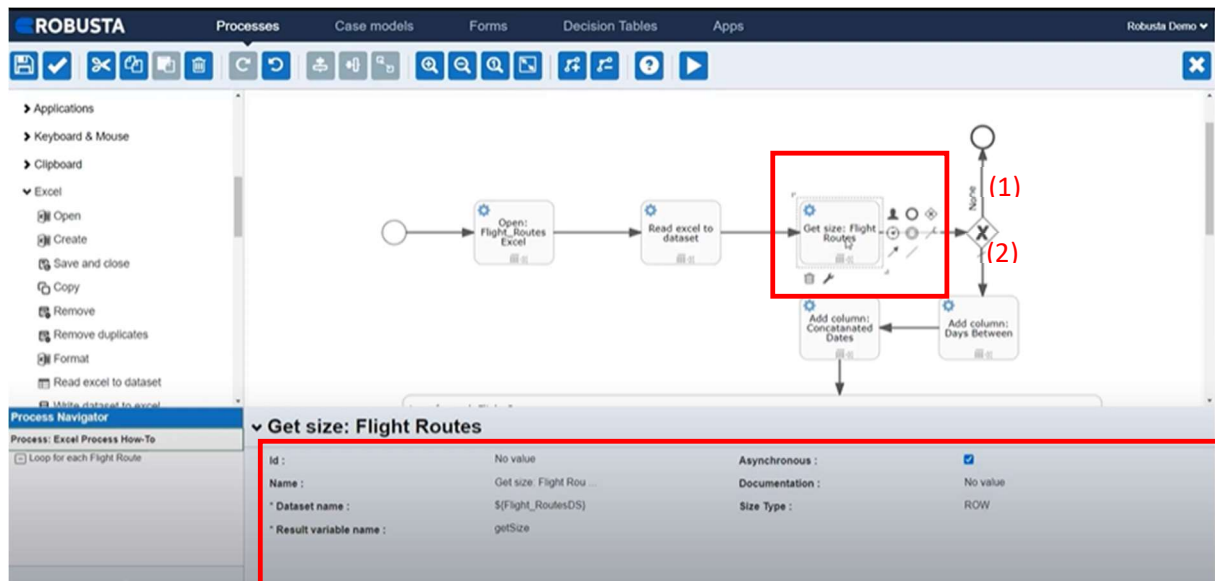
Excel > Read excel to dataset	
Name	Read excel to dataset
*Excel name	\${Flight_Routes}
Sheet name	Routes
*New dataset name	Flight_RoutesDS

- The next activity is Read Excel to Dataset activity, we transferred the data in the Excel file to a dataset. We checked the “Has header” box because our data has a header row. By doing so, it will be possible to specify the column header values in the Column fields in other activities at which we get or set cell values.

P.S1: If there is no header row in the table, we should not select the “Has header” option and we should specify which column to be processed in other activities with and index value of 0 1 2 or by entering the column header as A B C.

P.S2: If we left “*Sheet name*” field blank, the data on the Airports page, which is the first page, would be transferred to the dataset.

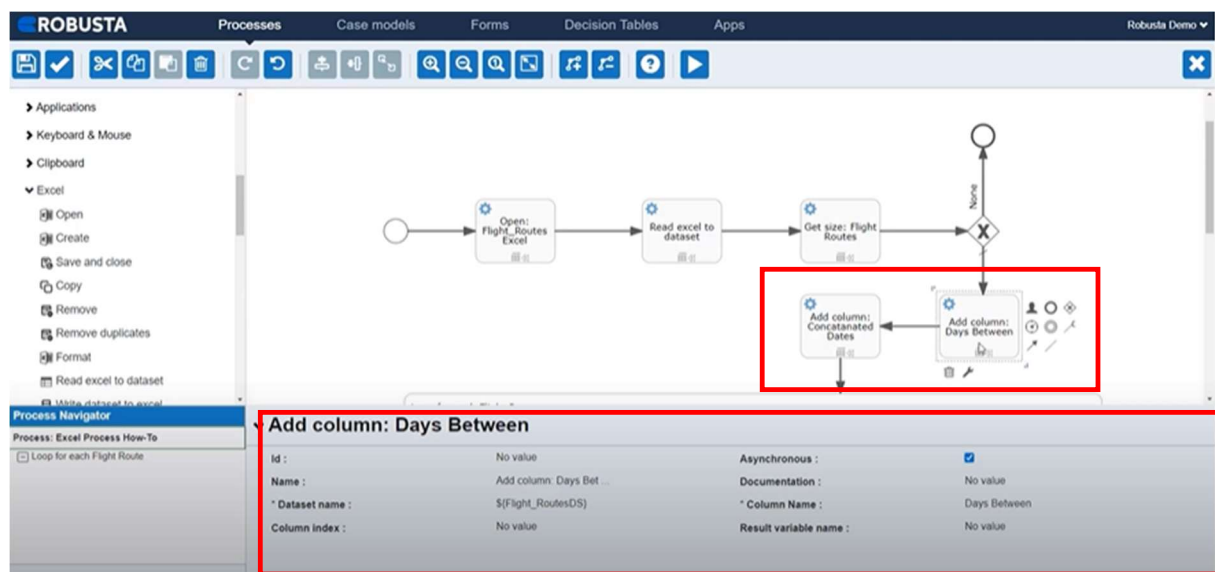
- In the next step, we need to get the row count of the dataset with the “Get Size” activity. After selecting the dataset from the list in the Dataset name field, we chose the “Row” option in the “Size Type” field because we want the number of rows to be counted. Then, in the “Result variable name” field, we defined the variable that we want to assign the row count value as ‘getSize’.



Dataset > Get size	
Name	Get size: Flight Routes
*Dataset name	\${Flight_RoutesDS}
*Result variable name	getSize
Size Type	ROW

- We have to ensure that the process needs to terminate if any row is not found in the excel file. To achieve this, we added an “Exclusive Gateway” that allows us to create alternative flows in our process. First (1) flow condition (  $\${getSize==0}$  ) checks whether the getSize variable value is equal to ‘true’ which means there is no row in the file. If this condition is met, the process is completed with an end event. In the second flow condition (2), we chose the “default flow” option checked ☒ and did not set any condition expression.

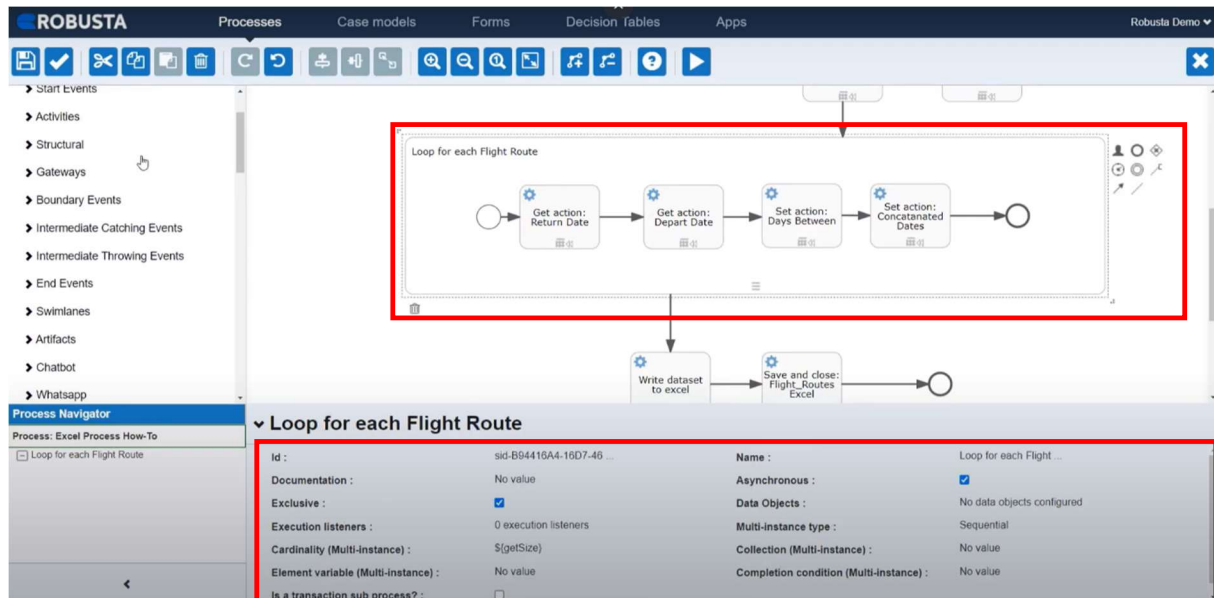
- We continue our process by adding two columns to the dataset with the “Add Column” activity. In this step, after selecting the dataset to add columns, we will write the column header name in the “Column Name” field. In the first activity we will add “*Days Between*” column after then “*Concatanated Days*” column. “*Days Between*” column is difference date of columns which are “*D.date*” and “*R.date*”. “*Concatanated Days*” column is concanate these two columns.



Dataset > Add column	
Name	Add column : Concacated Days
*Dataset name	\$(Flight_RoutesDS)
*Column name	Concatanated Days

Dataset > Add column	
Name	Add column : Days Between
*Dataset name	\$(Flight_RoutesDS)
*Column name	Days Between

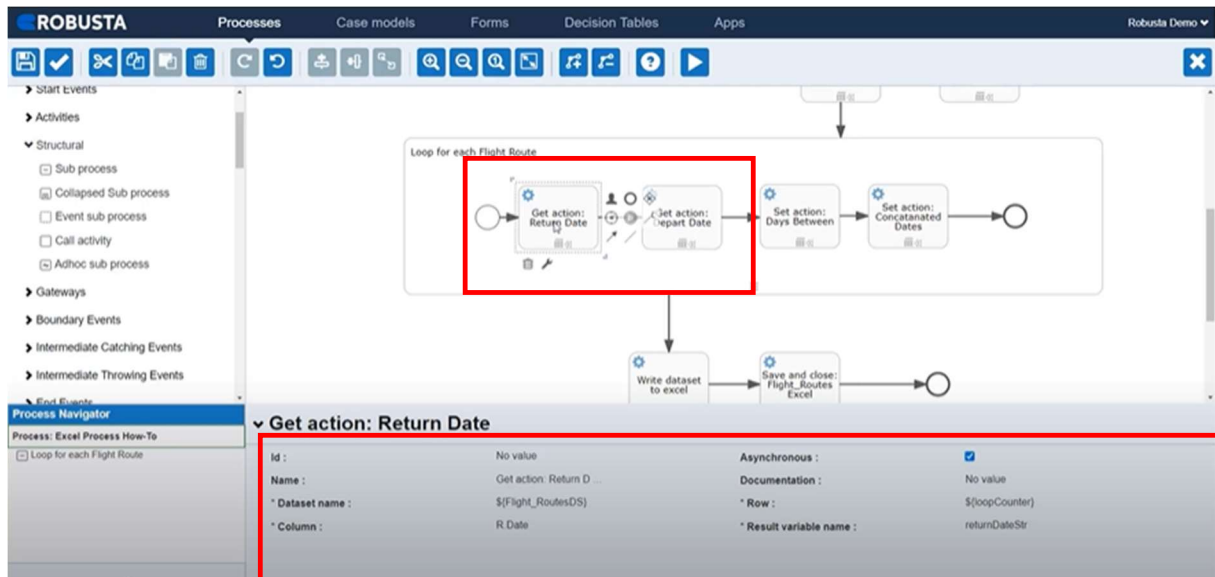
- The next activity is “Sub-Process” which is using here is recurring sub-process because we want to process each and every row in the dataset one by one.



Structural > Sub process	
Name	Loop for each flight route
Cardinality	\${getSize}
Multi-instance type	Sequential
Exclusive	<input checked="" type="checkbox"/> True

- We need to define how many times this sub-process will be repeated in the “Cardinality” field. This will be provided by using “*getSize*” variable.
- We chose the value of the “Multi-instance type” field as “Sequential” as we want each operation to be done sequentially in our sub-process loop. When we make these definitions, the “loopCounter” variable is automatically defined at the beginning of the loop. This variable which first takes the value 0, increases by 1 at each iteration. When the value of the “loopCounter” variable reaches to the Cardinality value, the loop is automatically terminated.

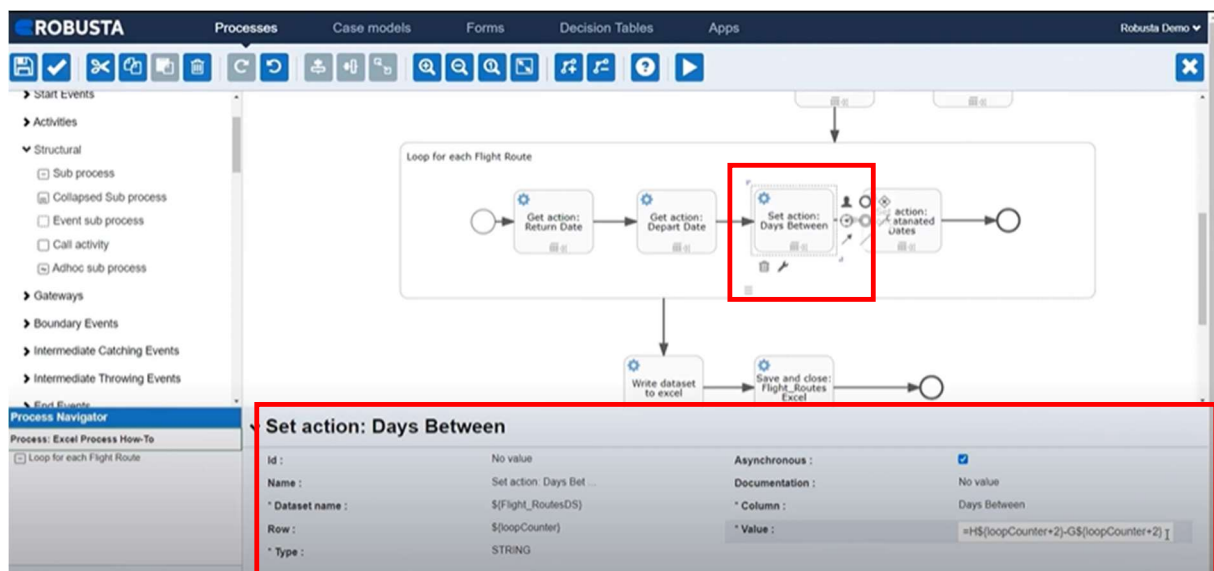
- In the sub-process, we first obtained the flight return and departure date from the dataset with the “Get action” activity. In this activity, we set the column header that we want to get data from in the “Column” field.



Dataset > Get action	
Name	Get action : Return Date
Dataset name	\$(Flight_RoutesDS)
*Column	R.Date
Result variable name	returnDateStr
Row	\$(loopCounter)

Dataset > Get action	
Name	Get action : Depart Date
Dataset name	\$(Flight_RoutesDS)
*Column	D.Date
Result variable name	departureDateStr
Row	\$(loopCounter)

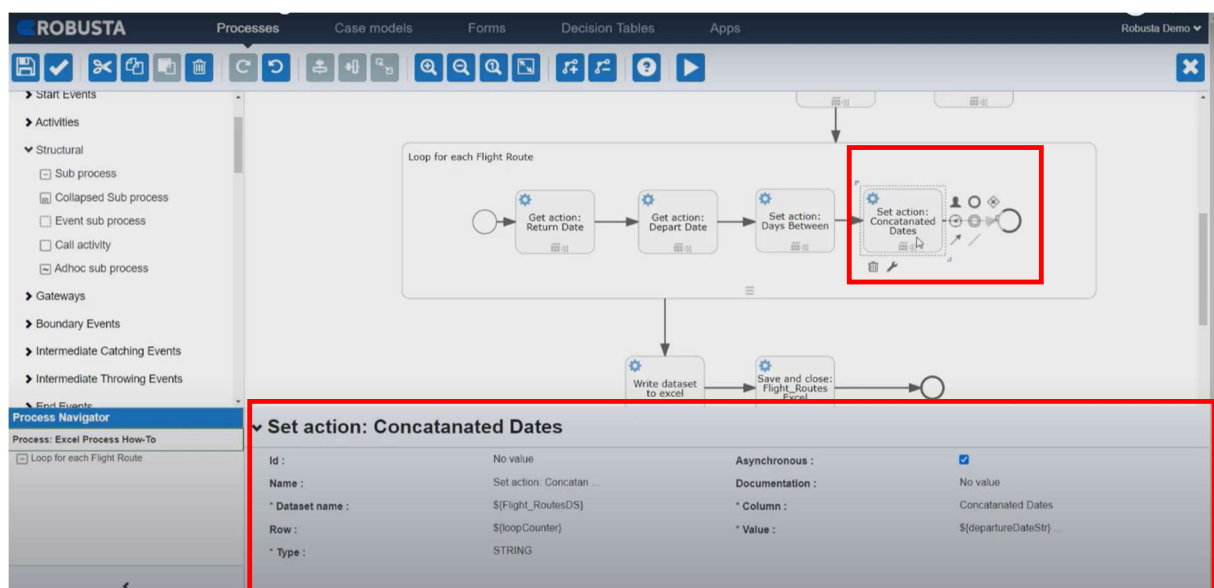
- Since we want to repeat the same operation for all the rows, we set 'loopCounter' variable in the Row field. The index value of rows and columns in the dataset starts with 0, same as loopCounter variable.
- Then, we will use the Set Action activity to set an excel formula to calculate the difference of the dates. Then we set to the newly created 'Days Between' column. In this activity, after selecting the dataset, we set the column header in the Column field and 'loopCounter' variable in the Row field. In the Value field, we wrote the formula( =H\${loopCounter+2}-G\${loopCounter+2} ) to calculate the difference between the dates by using the equal sign at the beginning.



Dataset > Set action	
Name	Set action : Days Between
Dataset name	\$(Flight_RoutesDS)
*Type	STRING
*Column	Days Between
*Value	=H\${loopCounter+2}-G\${loopCounter+2}
Row	\$(loopCounter)

- The dates we want to process start from the second row of the H and G columns in the table, the row value in the formula should also start from the 2<sup>nd</sup> row. So, in the formula syntax, we added 2 to the loopCounter variable.

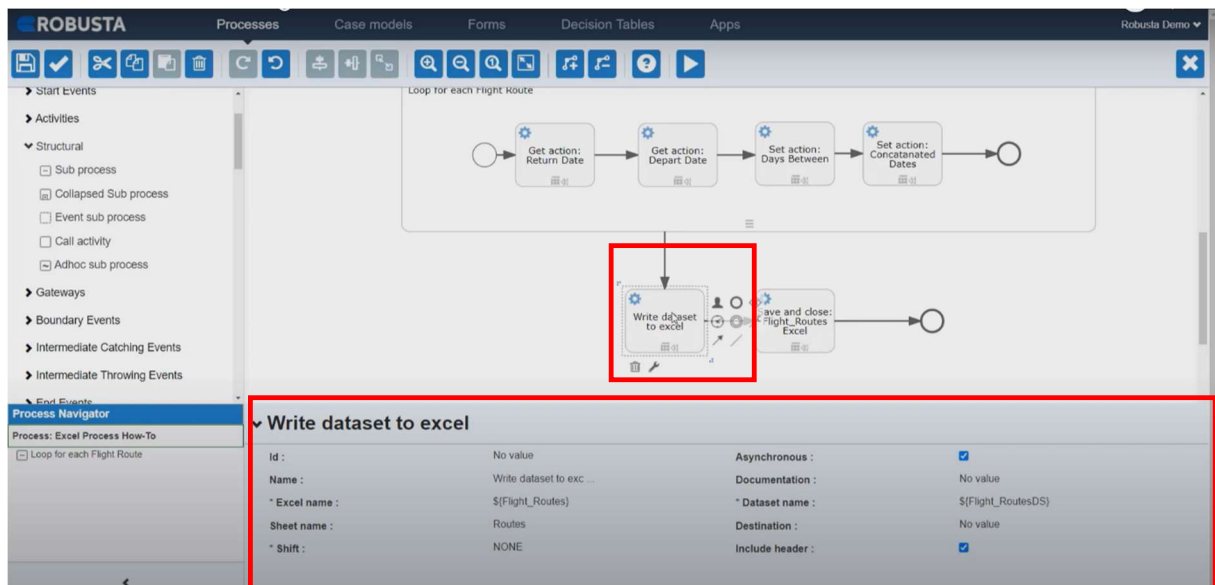
- In this step, we combined the dates and set the value in the newly added column named 'Concatanated Dates'. Here, we have written the variables that hold the date values in the Value field with a hyphen between them. Then we completed the loop activity with the End event, which allow us to end the process flow.



Dataset > Set action	
Name	Set action : Concatanated Dates
Dataset name	\${Flight_RoutesDS}
*Type	STRING
*Columns	Concatanated Dates
*Value	=\${departureDateStr}-\${returnDateStr}
Row	\${loopCounter}

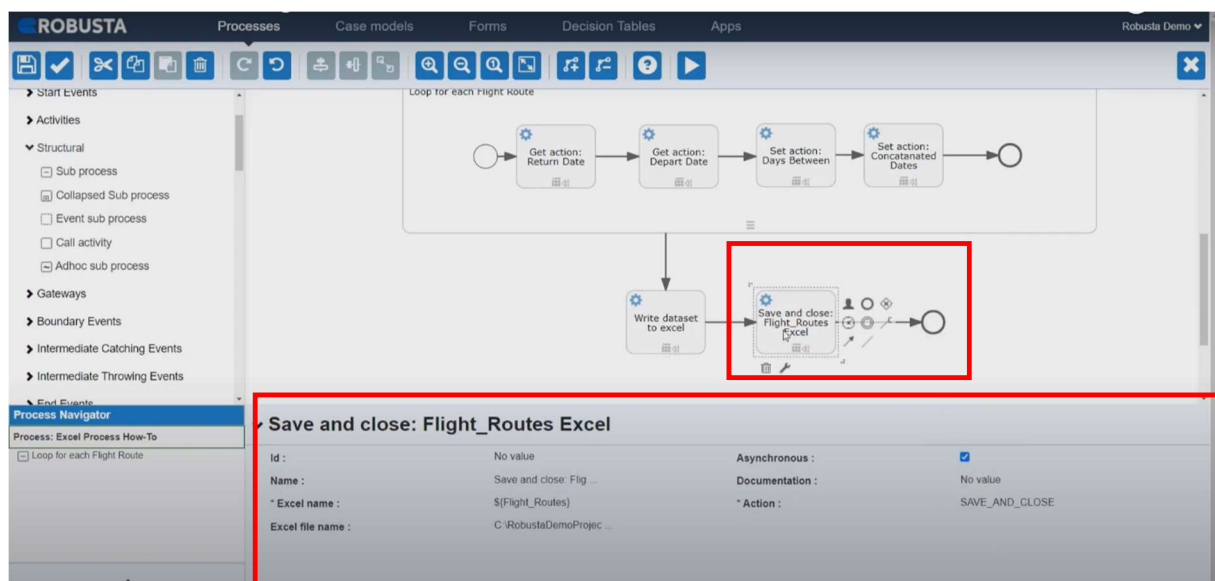


- We used Write Dataset to Excel activity after we finished the operations in the subprocess, which allows us to transfer all the data in a dataset to an Excel file. In this activity, after selecting the dataset and the Excel file, we chose the “Include header” option because we want to copy the header information for newly added columns.



Excel > Write data set to excel	
Name	Write dataset to Excel
*Dataset name	\${Flight_RoutesDS}
*Shift	NONE
*Excel name	\${Flight_Routes}
*Value	=\${departureDateStr}-\${returnDateStr}
Include header	<input checked="" type="checkbox"/> True
Sheet name	Routes

- We saved the Excel file in the output folder with the Save and Close activity. For this, we entered the directory, file name and extension of where we want to save our file in the Excel file name field. In the Action field, we selected the Save and Close option from the list and completed our process.



Excel > Write data set to excel

Name	Save and close : Flight_Routes_Excel
*Action	SAVE_AND_CLOSE
*Excel name	\${Flight_Routes}
*Excel file name	<a href="#">Current directory where the file is located</a>