PUBLIC

RPA As Easy As 1-2-3: Low-Code Design for Microsoft Office and SAP Software

INT165

Integration to MS Office Outlook

Xiaohui XUE, Moumita BERA / SAP

THE BEST RUN SAP

TABLE OF CONTENTS

INTRODUCTION

This exercise teach how Intelligent RPA can be used to interact with MS Outlook and be used to automatically process emails.

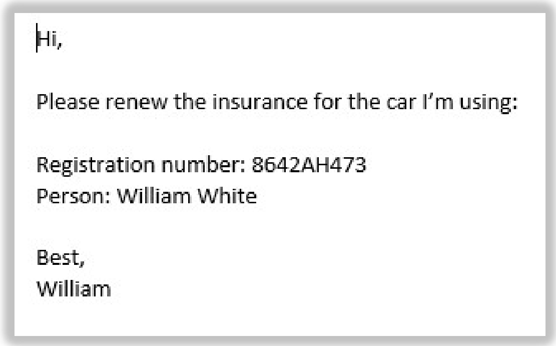
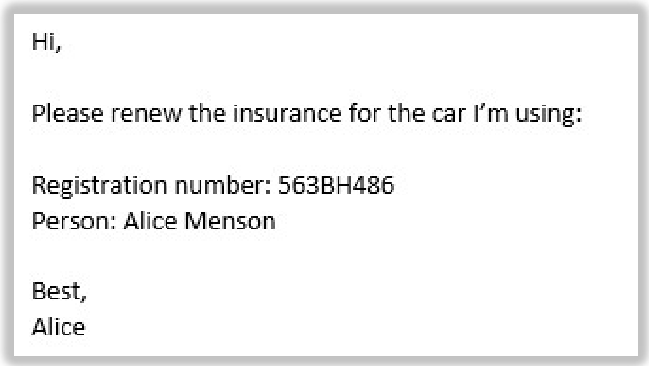
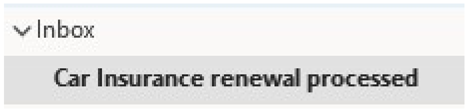
The use case involved in the exercise simulate an activity of a car fleet department in a company. The car fleet manages a pool of cars that are allocated to the company employees. Periodically they need to renew the car insurance.

For the car insurance renew process, each employee is requested to send an email to the car fleet department and indicate the car registration number. An email template is provided to all employees so all received requests have the same email subject and the same email format.

Then the car fleet department need to proceed these emails, group all similar requests, and forward to the insurance company.

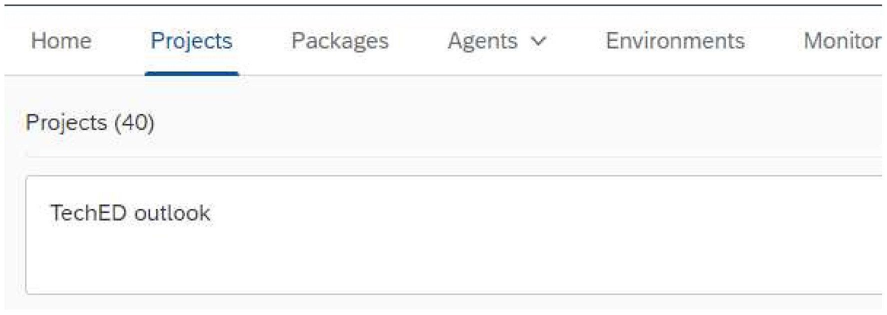
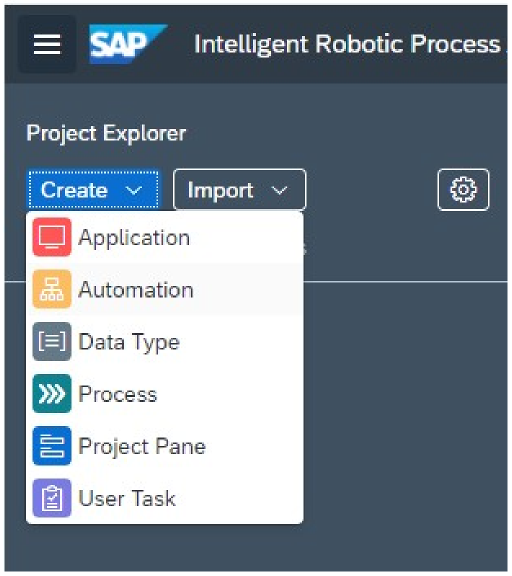
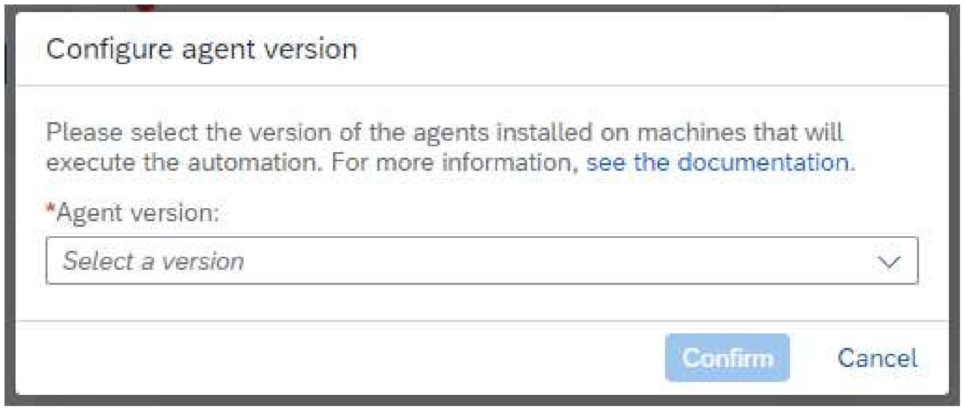
Estimated time: 30 minutes

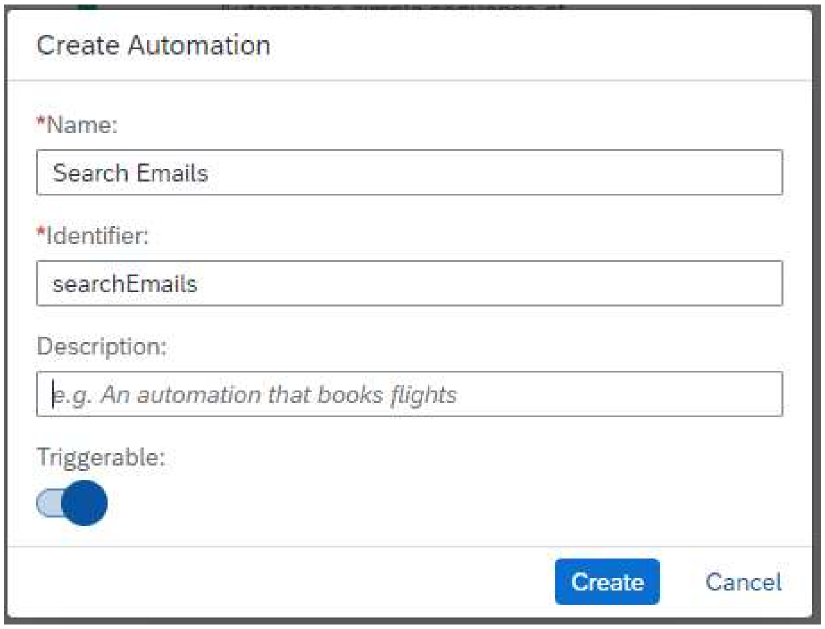
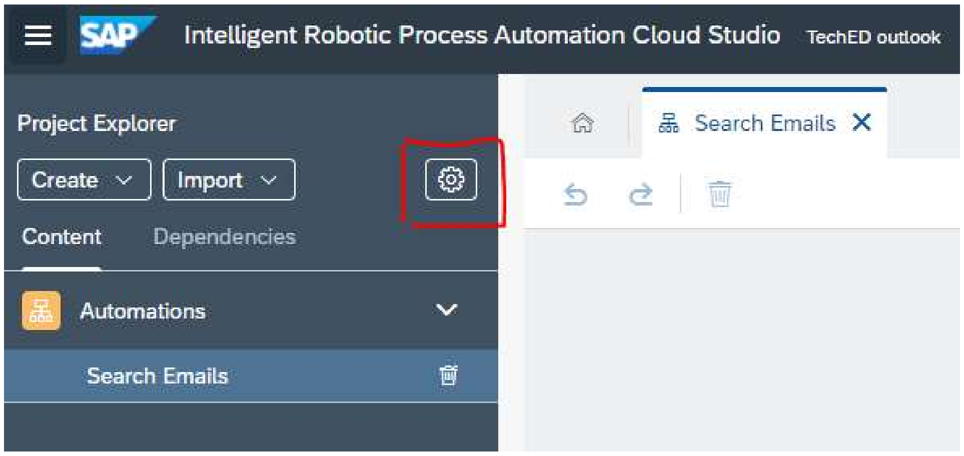
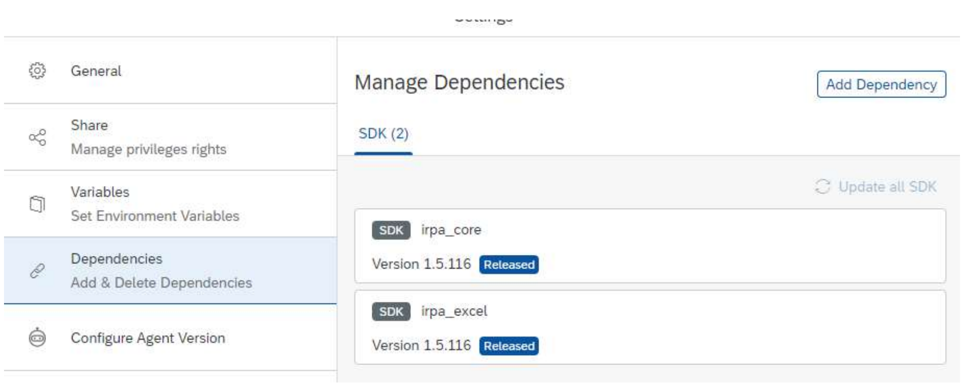
PREPARATION

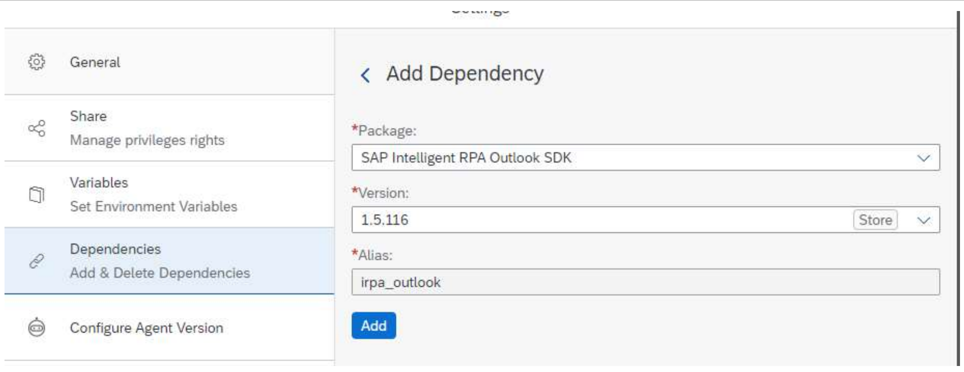
Explanation	Screenshot
Consider that the email account configured in the outlook used for this exercise represents the car fleet department. Send several emails to this account. All must have the subject "**Car Insurance Renewal**" and the body as: *Hi,* *Please renew the insurance for the car I'm using:* *Registration number: <SOME NUMBER>* *Person: <NAME>* *Best,* *William* Verify that the emails are all received in the outlook and make sure that they remain in the Inbox root folder.	 Hi, Please renew the insurance for the car I'm using: Registration number: 8642AH473 Person: William White Best, William  Hi, Please renew the insurance for the car I'm using: Registration number: 563BH486 Person: Alice Menson Best, Alice
Create a subfolder under the Inbox. The name of the subfolder is **Car Insurance renewal processed.**	

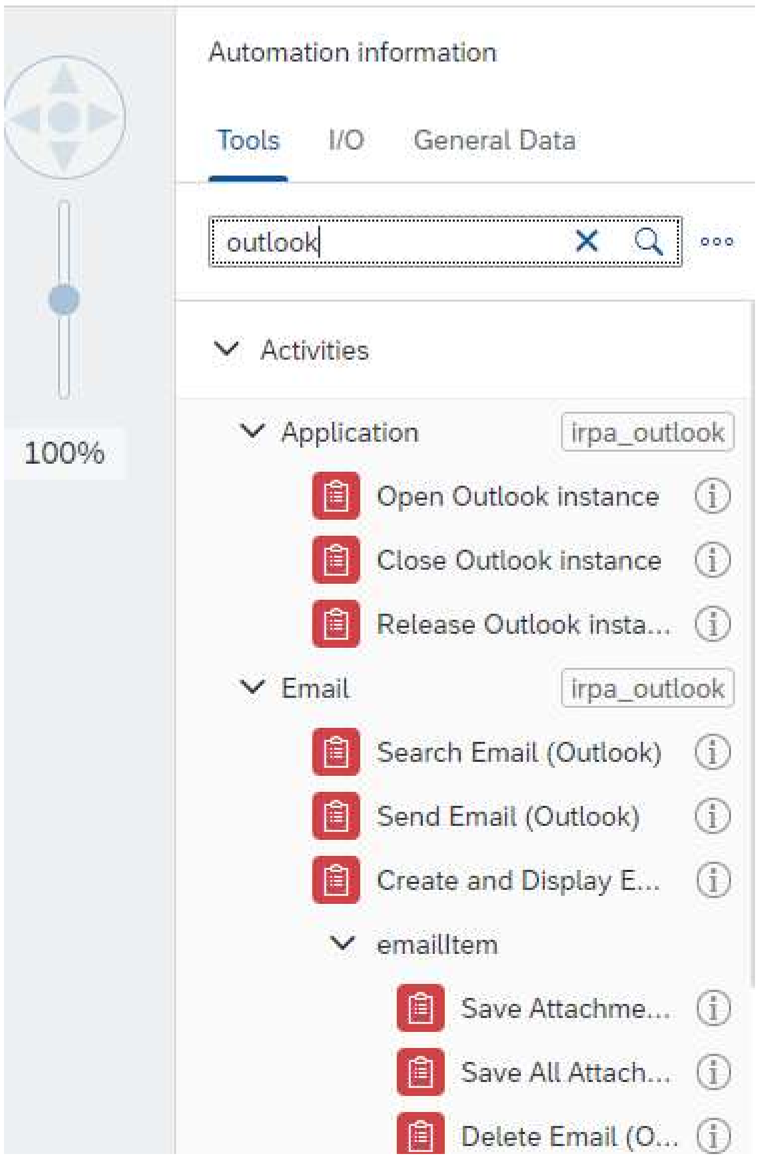
SEARCH EMAILS

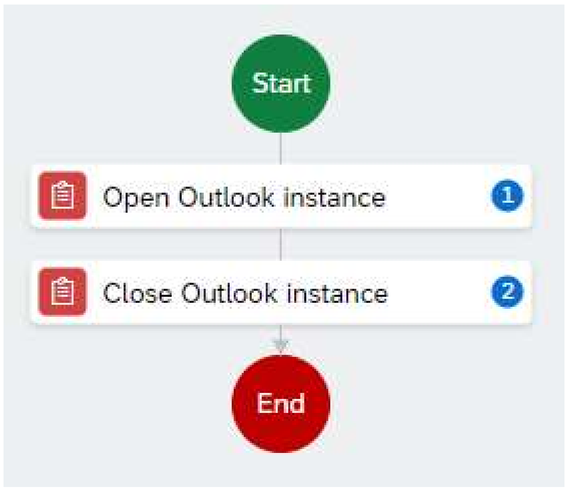
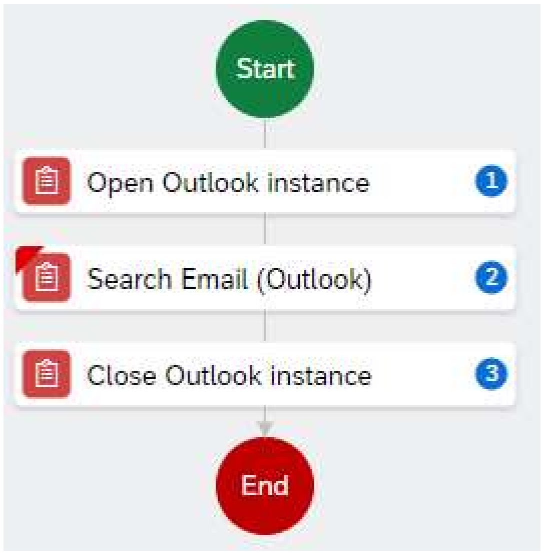
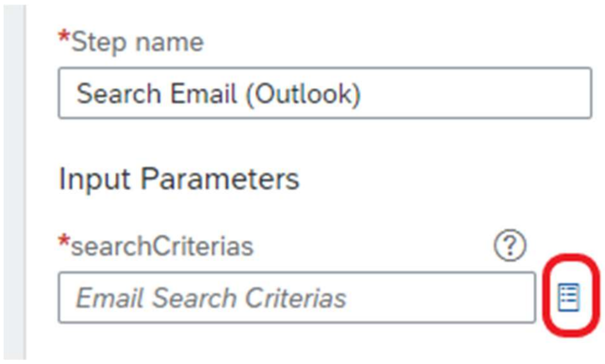
The first exercise shows how to search emails in outlook and navigate throw the result.

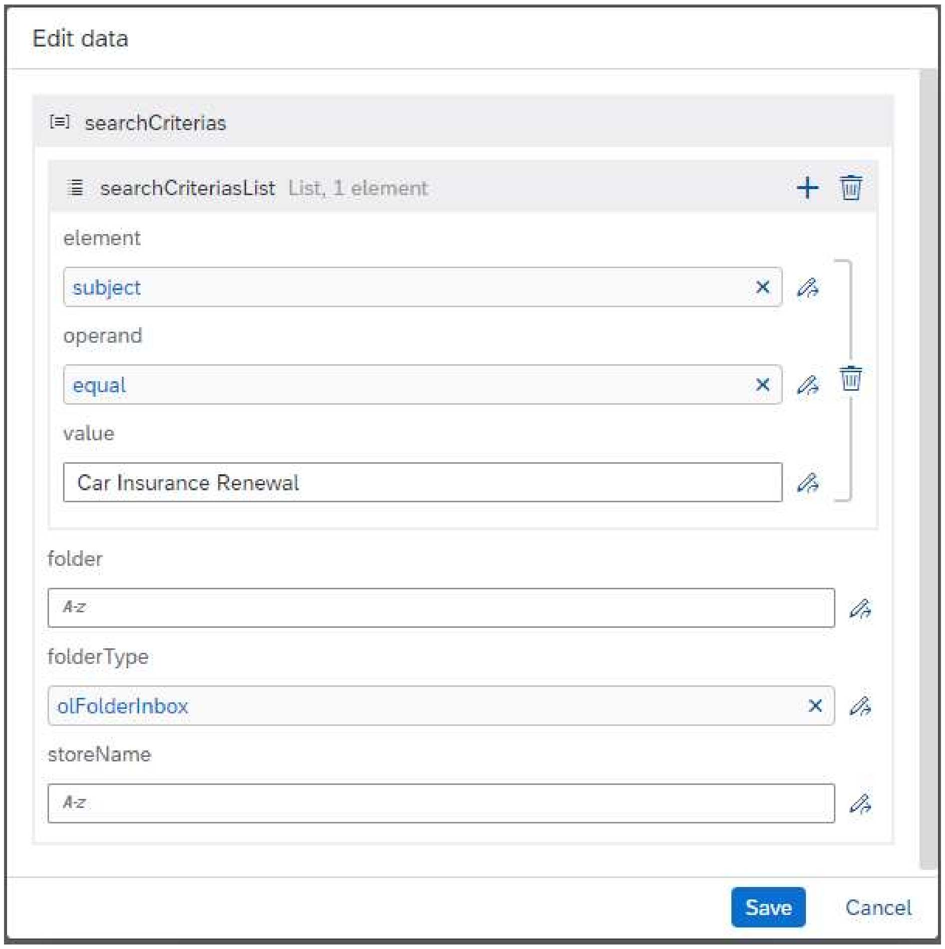
Explanation	Screenshot
Create a **project** in Intelligent RPA Factory, name it **TechED outlook**. The cloud studio will be opened in a separate tab.	 Home Projects Packages Agents Environments Monitor Projects (40) TechED outlook
In the project explorer, click on Create->Automation to create a new Automation.	 Intelligent Robotic Process Project Explorer Create Import Application Automation Data Type Process Project Pane User Task
In the pop-up dialog, select the **agent version** for this project. In this exercise, the version can be the one marked with the label **Local**. **Click** on **Confirm**.	Configure agent version Please select the version of the agents installed on machines that will execute the automation. For more information, see the documentation. *Agent version: Select a version

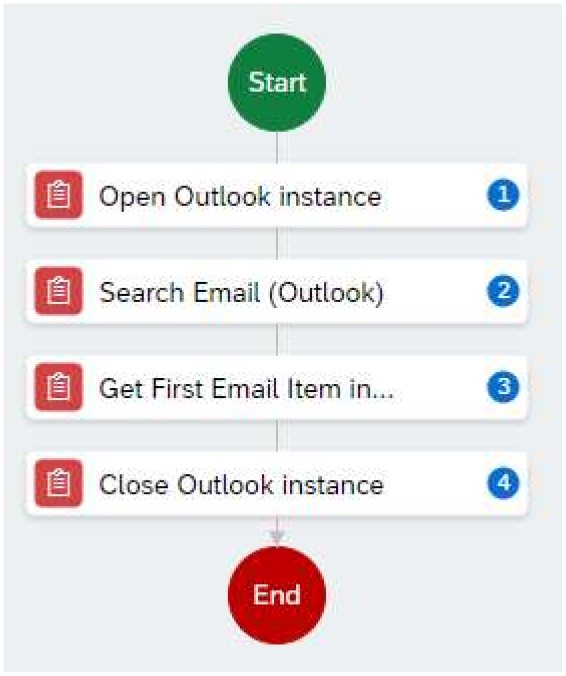
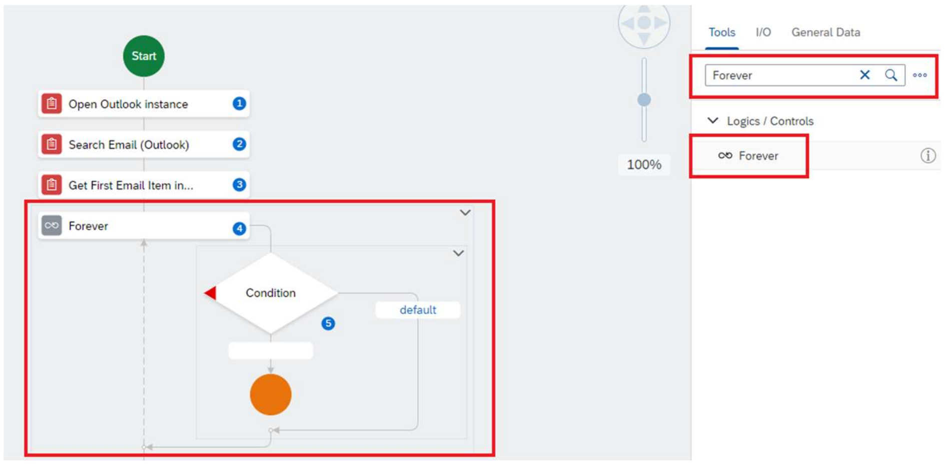
In the next dialog, put **Search Emails** as name for this automation. **Click** on **Create** to create the automation. This will create and open the Automation editor.	 Create Automation *Name: Search Emails *Identifier: searchEmails Description: e.g. An automation that books flights Triggerable:
Click on Project Settings icon and select the section **Dependencies**. Two dependencies should be automatically added which are irpa_core and irpa_excel. **Click** on **Add Dependency** icon.	  Manage Dependencies SDK irpa_core Version 1.5.116 Released SDK irpa_excel Version 1.5.116 Released

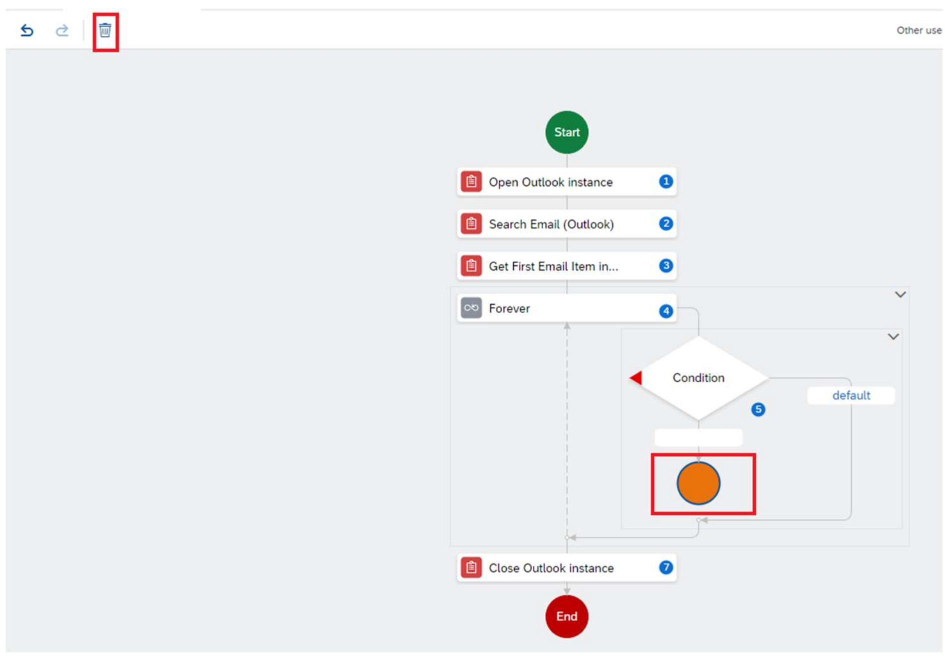
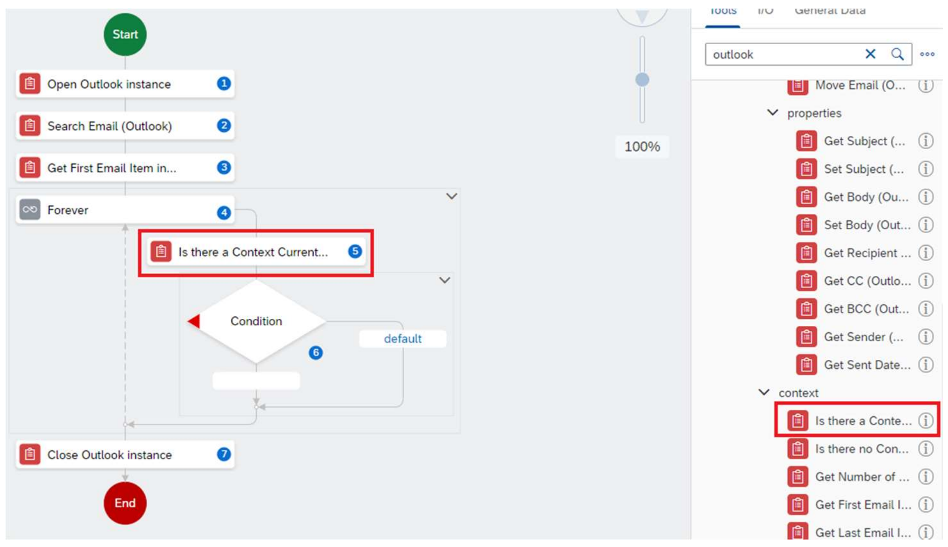
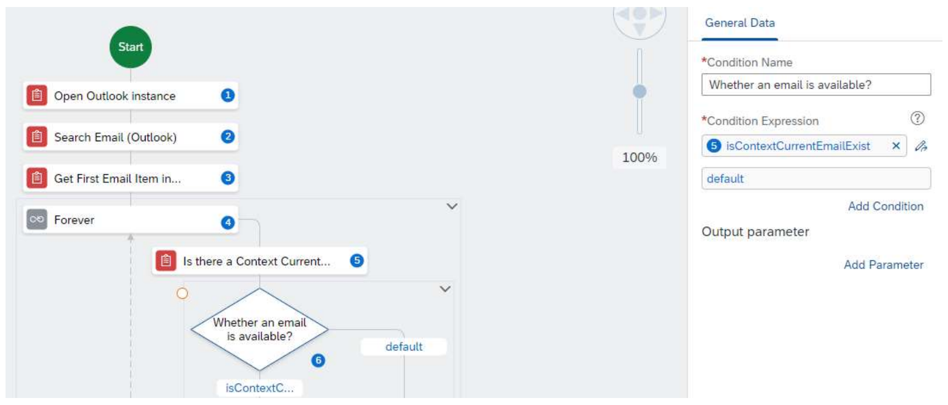
In the **Add Dependency** dialog, choose to add the package **SAP Intelligent RPA Outlook SDK**. Pick the latest version, regardless whether it's labeled with **Store**. **Click** on **Add** to add the dependency to the **TechED Outlook** project. **Close** the Project Settings dialog once it's done.	 General Share Manage privileges rights Variables Set Environment Variables Dependencies Add & Delete Dependencies Configure Agent Version Add Dependency Store Add *Package: SAP Intelligent RPA Outlook SDK *Version: 1.5.116 *Alias: irpa_outlook

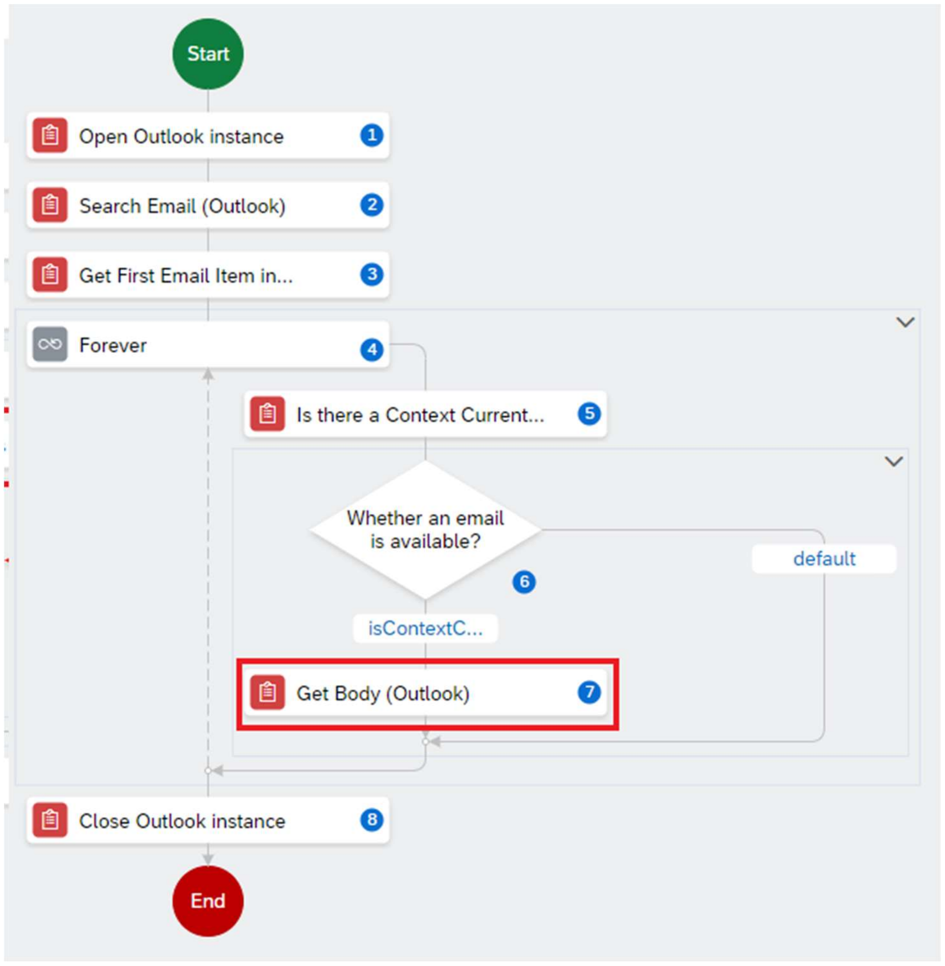
In the right panel of Automation editor, put **outlook** using the search bar. This will filter the activities that are related to outlook. Locate the two activities: • **Open Outlook instance** • **Close Outlook instance** Drag and drop them in the automation canvas between **Start** and **End** nodes. These 2 activities are mandatory for any use cases involving Outlook.	 100% Automation information Tools I/O General Data outlook Activities Application irpa_outlook Open Outlook instance Close Outlook instance Release Outlook insta... Email irpa_outlook Search Email (Outlook) Send Email (Outlook) Create and Display E... emailItem Save Attachme... Save All Attach... Delete Email (O...

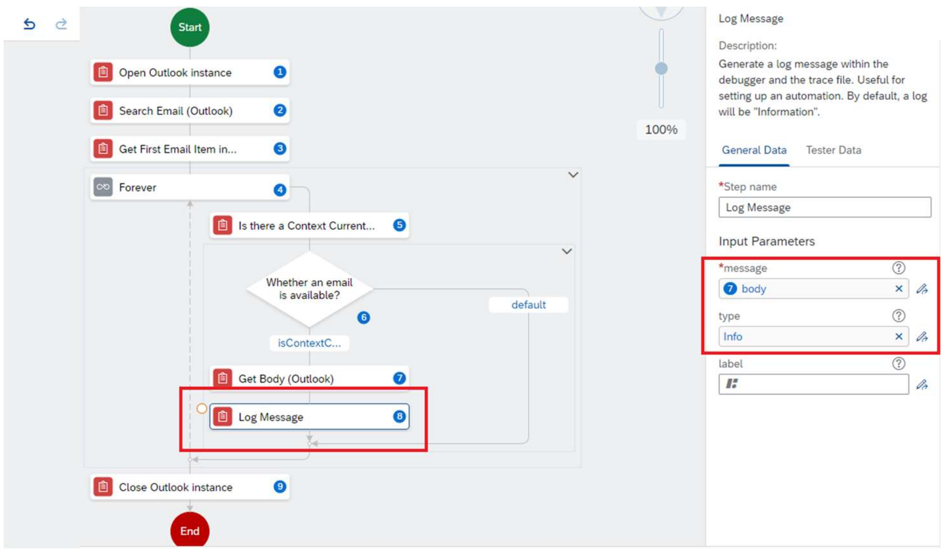
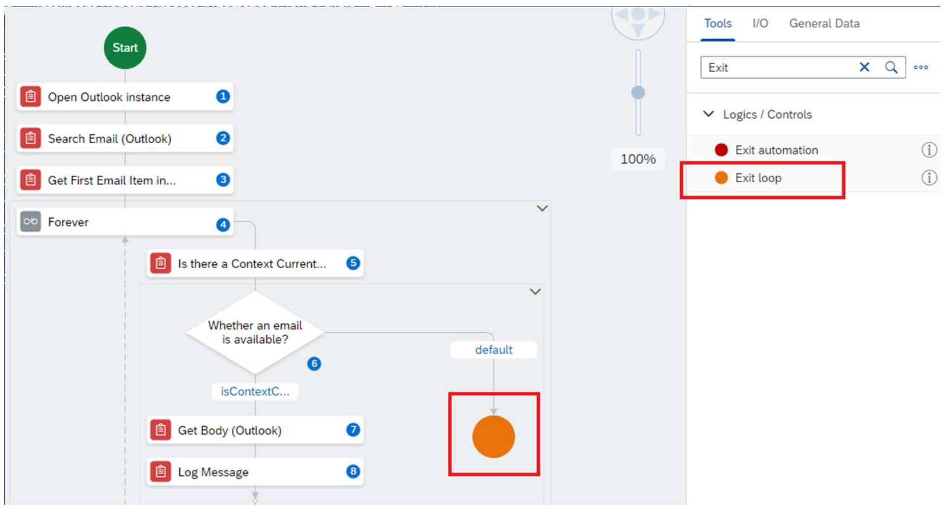
	 Start Open Outlook instance 1 Close Outlook instance 2 End
Drag and Drop the activity **Search Email (Outlook)** in the automation flow, in the middle between the steps **Open Outlook instance** and **Close Outlook instance**. Click on the step **Search Email (Outlook)** to edit its input. In the right panel, click on the expression editor icon for the input parameter **searchCriterias**.	 Start Open Outlook instance 1 Search Email (Outlook) 2 Close Outlook instance 3 End  *Step name Search Email (Outlook) Input Parameters *searchCriterias Email Search Criterias

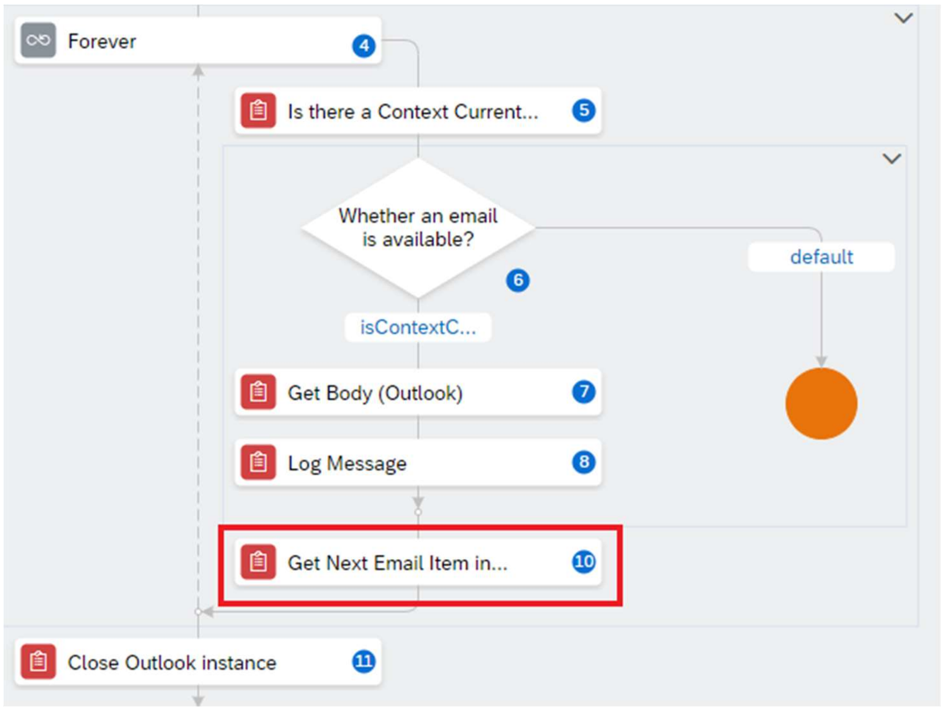
In the dialog, click on **+** icon to create a new search criterion and enter the below input. - **Element**: **subject** - **Operand: equal** - **Value: Car Insurance Renewal** In the button of the dialog, for the **folderType** parameter, select the value **olFolderInbox**. **Click** on **Save** to save the input parameter for email searching.	Edit data searchCriterias searchCriteriasList List, 1 element element: subject operand: equal value: Car Insurance Renewal folder: A-z folderType: olFolderInbox storeName: A-z

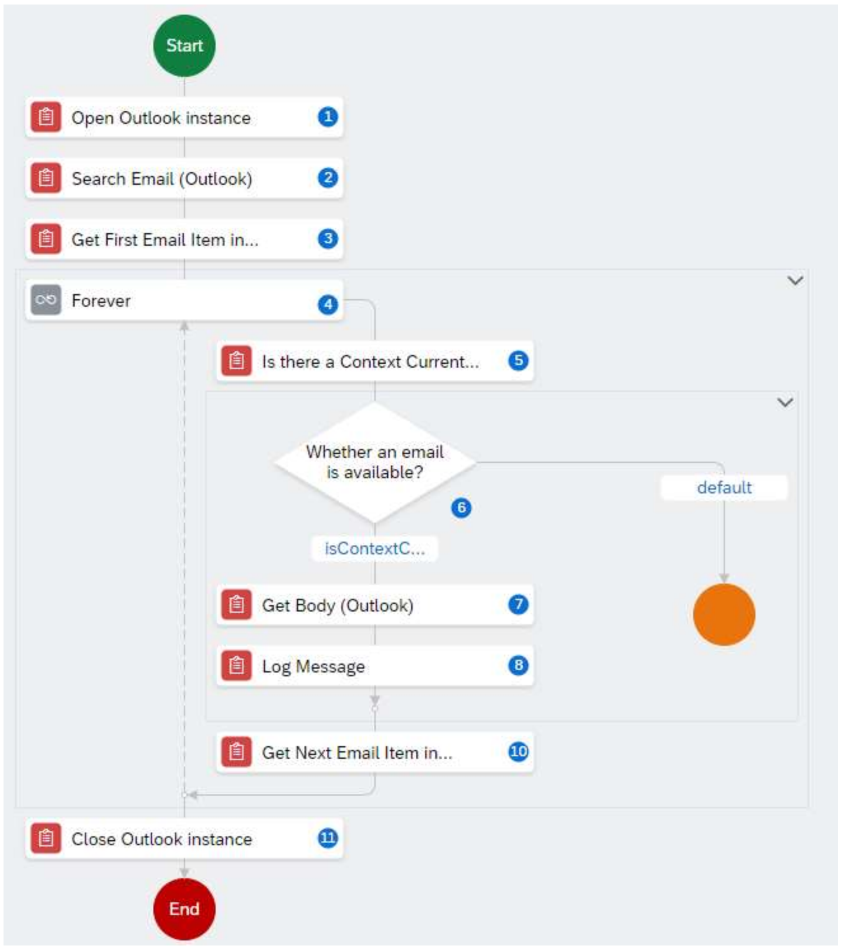
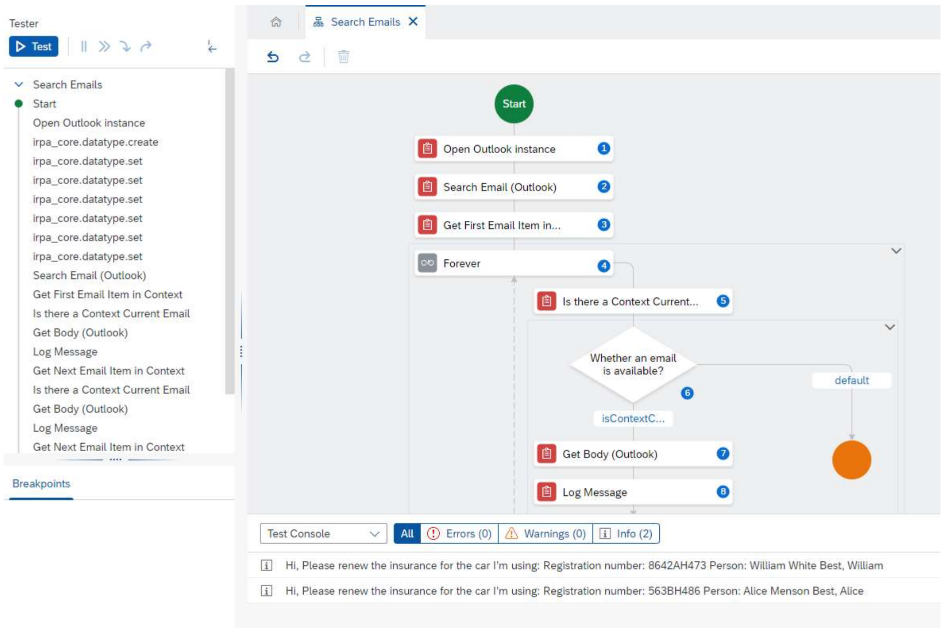
Drag and drop the activity **Get First Email Item in Context** in the automation. This activity allows to get the first email from the search result.	 Start Open Outlook instance 1 Search Email (Outlook) 2 Get First Email Item in... 3 Close Outlook instance 4 End
Search, and drag and drop the automation logic **Forever** in the automation. This will add a block automation flow with loop and condition.	 Start Open Outlook instance 1 Search Email (Outlook) 2 Get First Email Item in... 3 Forever 4 Condition default 5 100% Tools I/O General Data Forever Logics / Controls Forever

Select the Step node in the condition and click on delete button to remove it.	 Other use Start Open Outlook instance 1 Search Email (Outlook) 2 Get First Email Item in... 3 Forever 4 Condition 5 default Close Outlook instance 7 End
Drag and drop the activity **Is there a Context Current Email** before the **Condition** in the **Forever** loop.	 Start Open Outlook instance 1 Search Email (Outlook) 2 Get First Email Item in... 3 Forever 4 Is there a Context Current... 5 Condition 6 default Close Outlook instance 7 End 100% outlook Move Email (O... properties Get Subject (... Set Subject (... Get Body (Ou... Set Body (Out... Get Recipient ... Get CC (Outlo... Get BCC (Out... Get Sender (... Get Sent Date... context Is there a Conte... Is there no Con... Get Number of ... Get First Email I... Get Last Email I...
Click on the **Condition** node to edit it. In the right panel, edit the **Condition name** to **Whether an email is available**? In the **ConditionExpression**, put the output from the	 Start Open Outlook instance 1 Search Email (Outlook) 2 Get First Email Item in... 3 Forever 4 Is there a Context Current... 5 Whether an email is available? 6 default isContextC... 100% General Data Add Condition Output parameter Add Parameter *Condition Name: Whether an email is available? *Condition Expression: 5 isContextCurrentEmailExist default

previous step **isContextCurrentEmailExist.**	
Search, drag and drop the activity GetBody (Outlook) in the condition branch where the condition is satisfied.	 Start Open Outlook instance 1 Search Email (Outlook) 2 Get First Email Item in... 3 Forever 4 Is there a Context Current... 5 Whether an email is available? 6 default isContextC... Get Body (Outlook) 7 Close Outlook instance 8 End

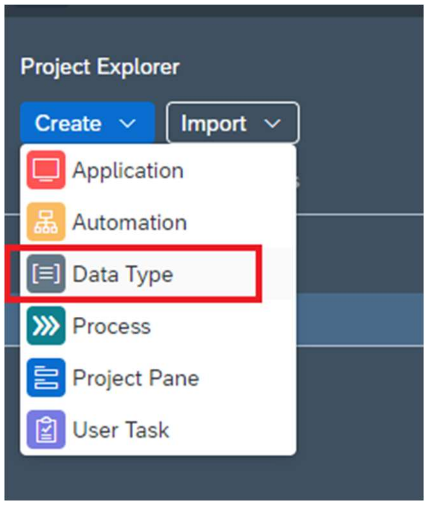
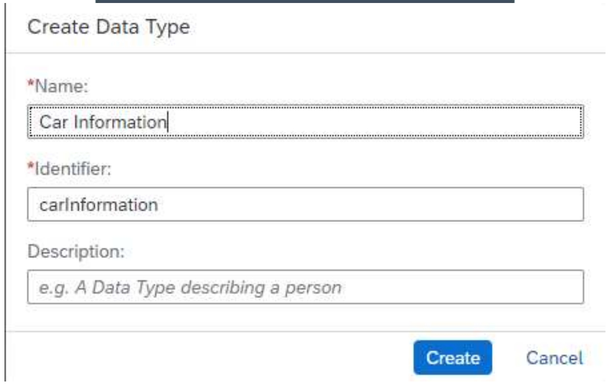
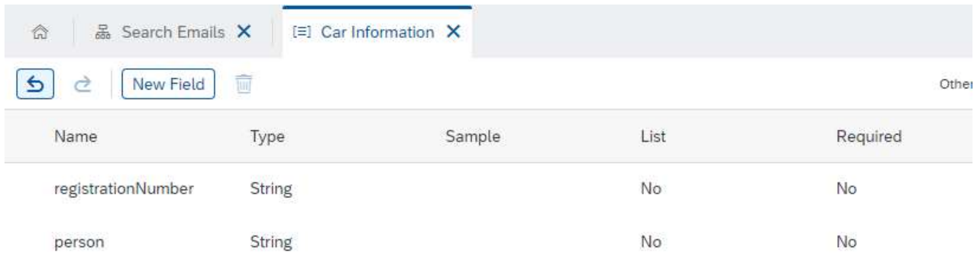
Search, drag and drop the activity **Log Message**. Edit this step to put the **body** output from the Step 7 in **message** parameter. The step parameter **type** is **Info**.	 Start Open Outlook instance 1 Search Email (Outlook) 2 Get First Email Item in... 3 Forever 4 Is there a Context Current... 5 Whether an email is available? 6 default isContextC... Get Body (Outlook) 7 Log Message 8 Close Outlook instance 9 End 100% General Data Tester DataLog Message Description: Generate a log message within the debugger and the trace file. Useful for setting up an automation. By default, a log will be "Information". *Step name: Log Message Input Parameters *message: 7 body type: Info label:
Search, drag and drop the control element **Exit loop** in the condition **default** branch. The objective is to exist the loop when there's no more email to process.	 Start Open Outlook instance 1 Search Email (Outlook) 2 Get First Email Item in... 3 Forever 4 Is there a Context Current... 5 Whether an email is available? 6 default isContextC... Get Body (Outlook) 7 Log Message 8 100% Tools I/O General Data Exit Logics / Controls Exit automation Exit loop

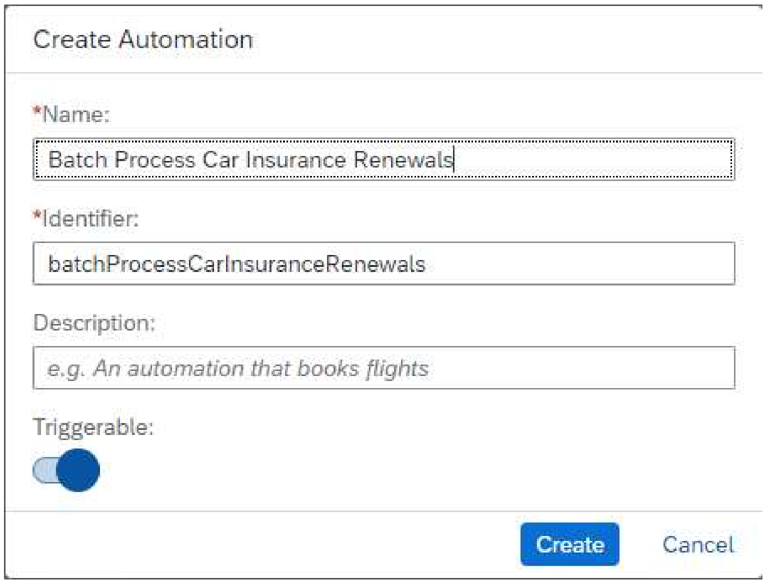
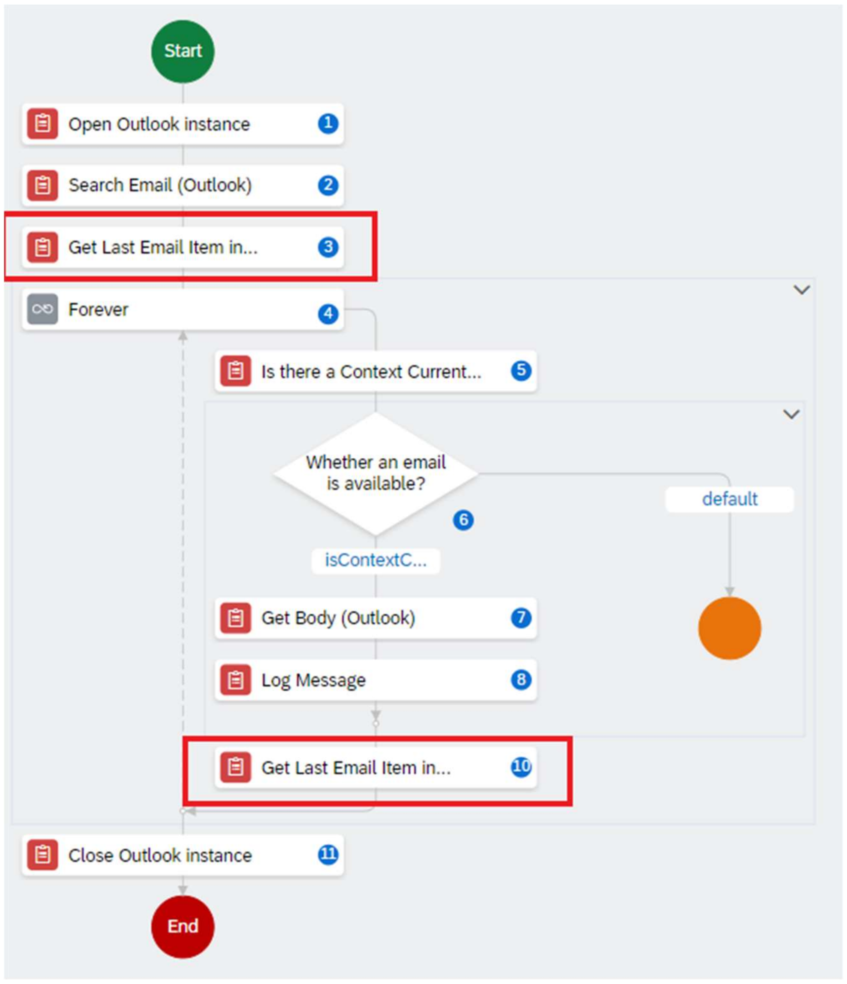
Search, drag and drop the activity **Get Next Email Item in Context** in the automation, in the loop, after the condition block.	

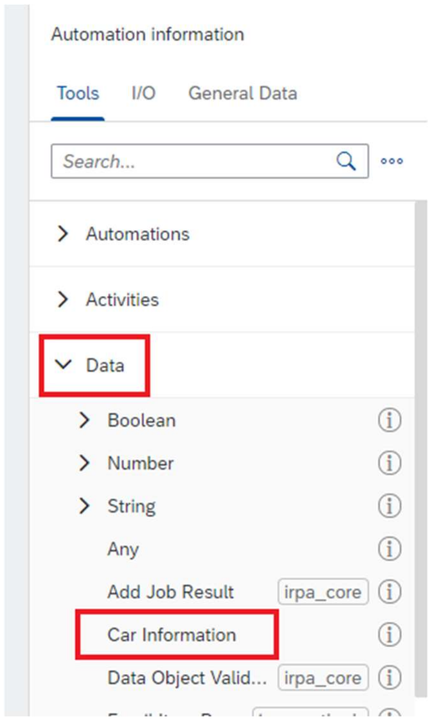
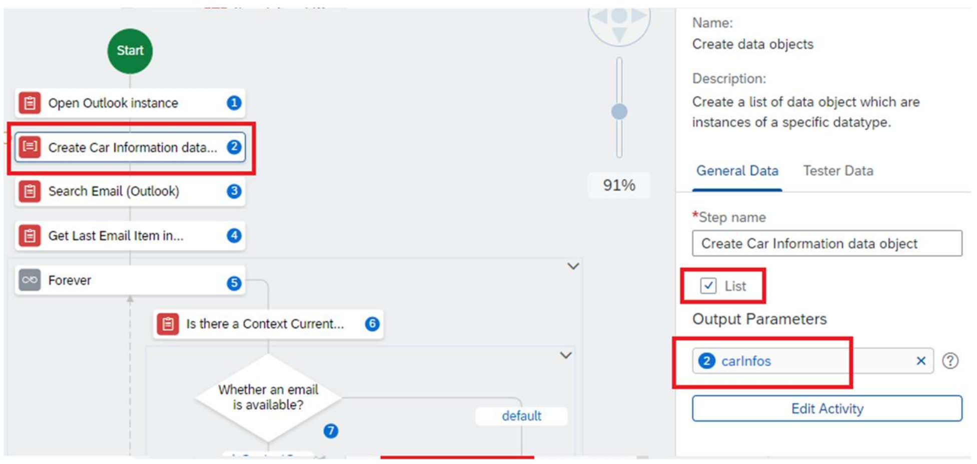
At the end, the automation should look at this. **Save** the automation using the **Save button** on the top right corner of the Automation editor.	 Start Open Outlook instance 1 Search Email (Outlook) 2 Get First Email Item in... 3 Forever 4 Is there a Context Current... 5 Whether an email is available? 6 default isContextC... Get Body (Outlook) 7 Log Message 8 Get Next Email Item in... 10 Close Outlook instance 11 End
Test your automation by clicking on the Test button. Configure the environment in the pop-up dialog. Wait the automation to be built, download and executed. And the tester will be opened. The test execution also logs the email bodies in the **Test Console** as **Info**.	 Tester Test Search Emails Breakpoints Start Open Outlook instance 1 Search Email (Outlook) 2 Get First Email Item in... 3 Forever 4 Is there a Context Current... 5 Whether an email is available? 6 default isContextC... Get Body (Outlook) 7 Log Message 8 Test Console All Errors (0) Warnings (0) Info (2) Search Emails Start Open Outlook instance irpa_core.datatype.create irpa_core.datatype.set irpa_core.datatype.set irpa_core.datatype.set irpa_core.datatype.set irpa_core.datatype.set irpa_core.datatype.set Search Email (Outlook) Get First Email Item in Context Is there a Context Current Email Get Body (Outlook) Log Message Get Next Email Item in Context Is there a Context Current Email Get Body (Outlook) Log Message Get Next Email Item in Context Hi, Please renew the insurance for the car I'm using: Registration number: 8642AH473 Person: William White Best, William Hi, Please renew the insurance for the car I'm using: Registration number: 563BH486 Person: Alice Menson Best, Alice

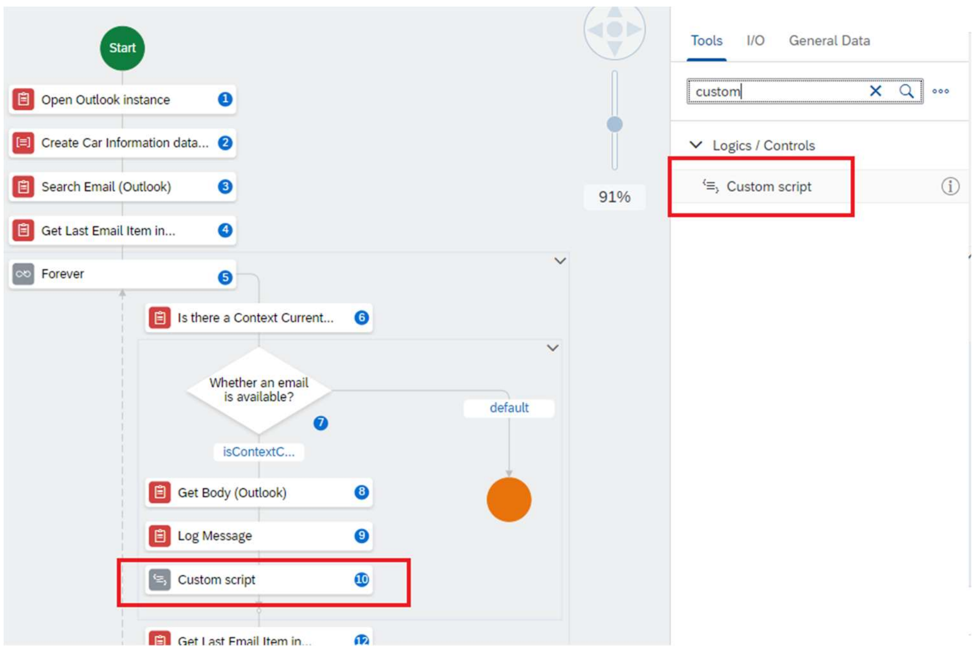
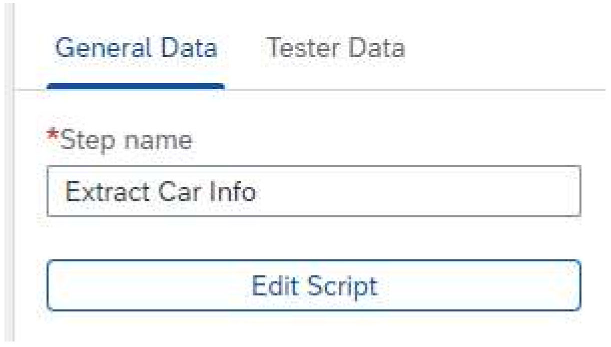
BATCH PROCESS INSURANCE RENEWAL AND SEND EMAIL

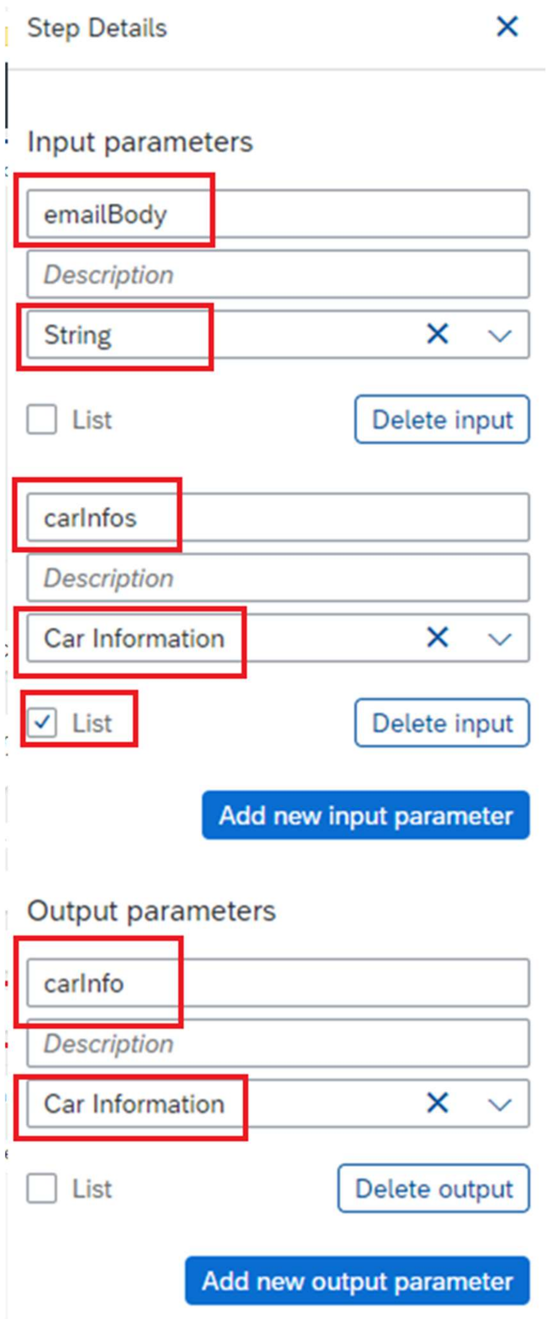
This exercise shows an advanced use case that extracts the email content, process them and send an grouped insurance renew request.

Explanation	Screenshot
Create a **Data Type** in the project. In the pop-up dialog, put **Car Information** as **Name**. **Click** on **Create**. This will open the data type editor.	 Project Explorer Create Import Application Automation Data Type Process Project Pane User Task Create Data Type *Name: Car Information *Identifier: carInformation Description: e.g. A Data Type describing a person
Click on **New Field** 2 times to create two fields: - **registrationNumber** - **person** Both are of **String** type. **Save** the data type.	 Name / Type / Sample / List / Required registrationNumber / String / / No / No person / String / / No / No

Now, create a new automation again. Name it **Batch Process Car Insurance Renewals**. **Click** on **Create** to create the Automation.	 Create Automation *Name: Batch Process Car Insurance Renewals *Identifier: batchProcessCarInsuranceRenewals Description: *e.g. An automation that books flights* Triggerable: Create Cancel
Create an automation that is similar from the previous section. But this time, uses the activity **Get Last Email Item in Context** to replace: • Get First Email Item in Context • Get Next Email Item in Context Don't run or test the automation until you add the activity **Move Email (Outlook)**!	 Start Open Outlook instance 1 Search Email (Outlook) 2 Get Last Email Item in... 3 Forever 4 Is there a Context Current... 5 Whether an email is available? 6 default isContextC... Get Body (Outlook) 7 Log Message 8 Get Last Email Item in... 10 Close Outlook instance 11 End

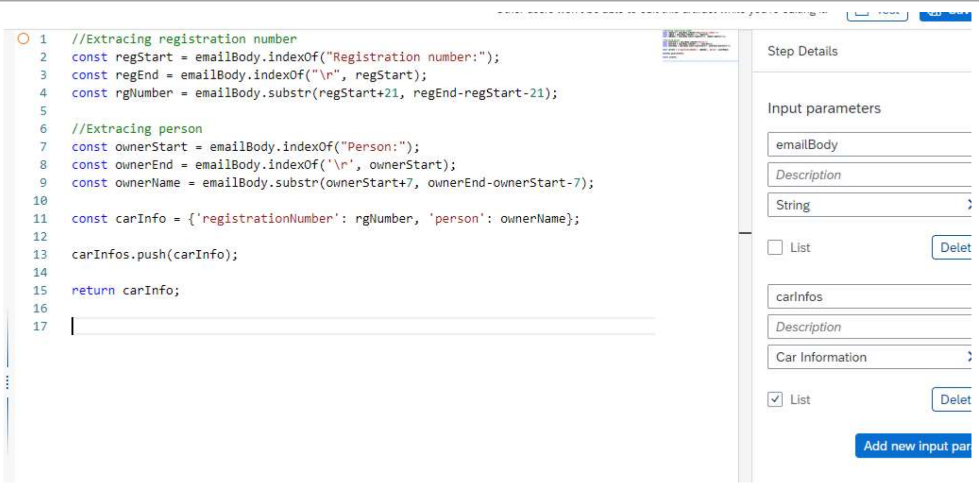
Locate in the right panel, the data type **Car Information** previously created in the project.	 Automation information; Tools, I/O, General Data; Search...; Automations; Activities; Data; Boolean; Number; String; Any; Add Job Result (irpa_core); Car Information; Data Object Valid... (irpa_core)
Drag and drop this data type in the automation before the step Search Email. This means the automation will create and instance of this data type. Click on this step to open its configuration: • Click on the checkbox **List** to instantiate a list of **Car Information.** • Edit Output Parameters to rename the output as carInfos	 Name: Create data objects Description: Create a list of data object which are instances of a specific datatype. General Data / Tester Data *Step name: Create Car Information data object ☑ List Output Parameters: 2 carInfos Edit Activity Flow: Start; 1 Open Outlook instance; 2 Create Car Information data...; 3 Search Email (Outlook); 4 Get Last Email Item in...; 5 Forever; 6 Is there a Context Current...; 7 Whether an email is available?; default; 91%

Search for **Custom Script** in **Tools**. Drag and drop **Custom Script** in the automation after the **Log Message** step. It should remain in the condition branch where there's an email to process. Click on the **Custom Script** to edit it.	 Start Open Outlook instance 1 Create Car Information data... 2 Search Email (Outlook) 3 Get Last Email Item in... 4 Forever 5 Is there a Context Current... 6 Whether an email is available? 7 default isContextC... Get Body (Outlook) 8 Log Message 9 Custom script 10 Get Last Email Item in... 12 91% Tools I/O General Data custom Logics / Controls Custom script
In the right panel, put **Extract Car Info** as **Step name**; Click on **Edit Script** to edit the custom script.	 *Step name: Extract Car Info

In the opened custom script editor, edit the step details to add 2 input parameters and 1 output parameter: • Input: **emailBody**, of type **String** • Input: **carInfos**, of type list of **Car Information** • Output, **carInfo**, of type **Car Information**	 Step Details Input parameters emailBody Description String List carInfos Description Car Information ✓ List Output parameters carInfo Description Car Information List Delete input Delete input Add new input parameter Delete output Add new output parameter

In the code editor, copy-past the code located in the file ***[INT165] Custom Script - Extract Car Info.txt***

This piece of code extracts the registration number and person for each car and add it in the list of carInfos. It also returns the extracted car information as output.



```
//Extracing registration number
const regStart = emailBody.indexOf("Registration number:");
const regEnd = emailBody.indexOf("\r", regStart);
const rgNumber = emailBody.substr(regStart+21, regEnd-regStart-21);

//Extracing person
const ownerStart = emailBody.indexOf("Person:");
const ownerEnd = emailBody.indexOf('\r', ownerStart);
const ownerName = emailBody.substr(ownerStart+7, ownerEnd-ownerStart-7);

const carInfo = {'registrationNumber': rgNumber, 'person': ownerName};

carInfos.push(carInfo);

return carInfo;

```

Step Details

Input parameters

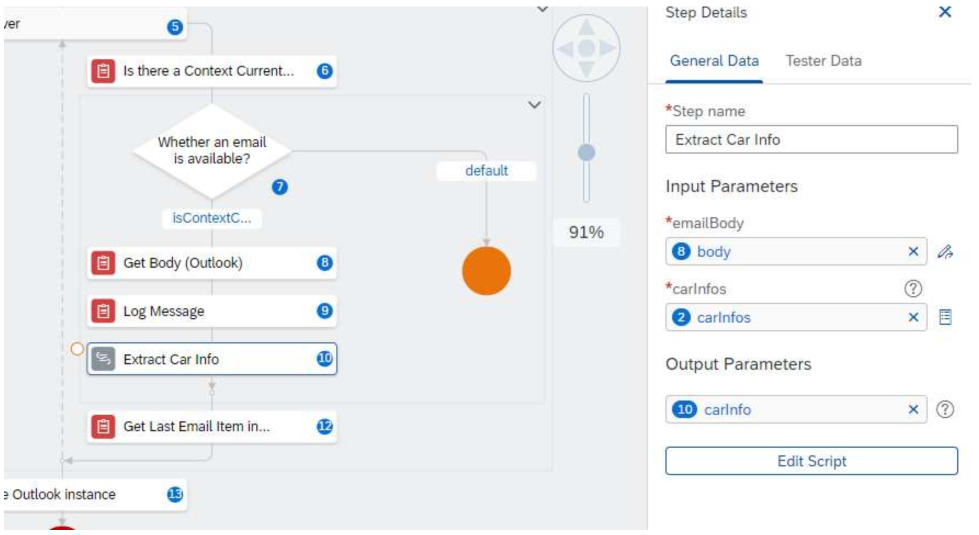
emailBody
Description
String
List

carInfos
Description
Car Information
✓ List

Go back to the automation and click again on the step Extract Car Info.

Bind the input parameters as follows:

- Input: **emailBody** takes the output of the **step 8**
- Input: carInfos, takes the output of the **Step 2** which is the list of **Car Information**



Step Details

General Data | Tester Data

*Step name
Extract Car Info

Input Parameters

*emailBody
8 body

*carInfos
2 carInfos

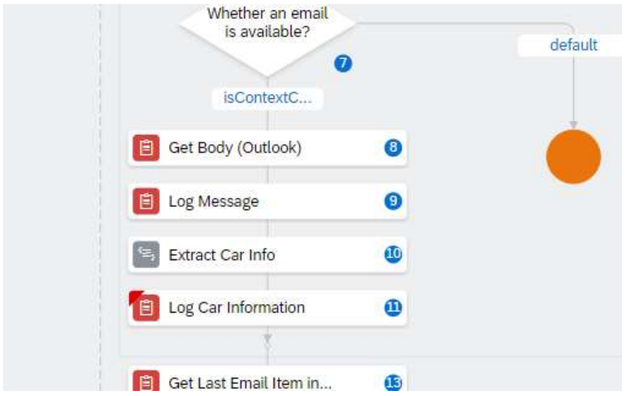
Output Parameters

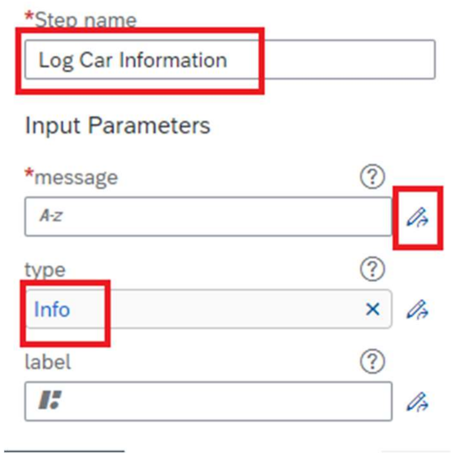
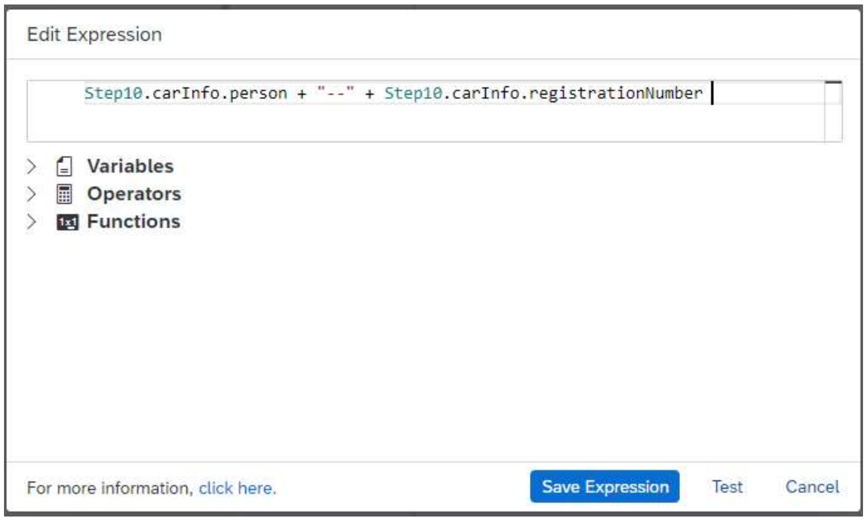
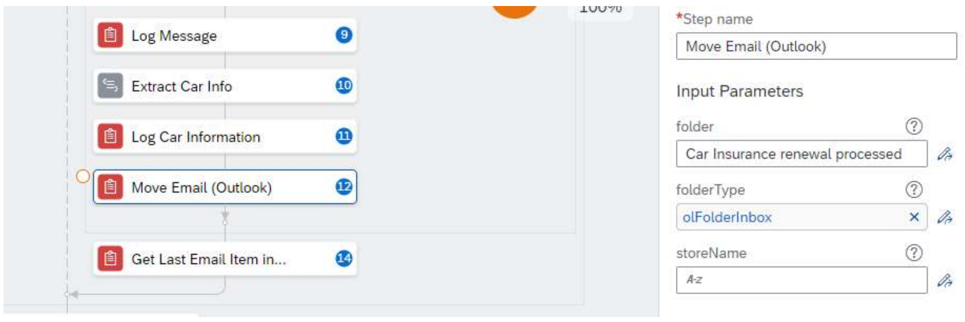
10 carInfo

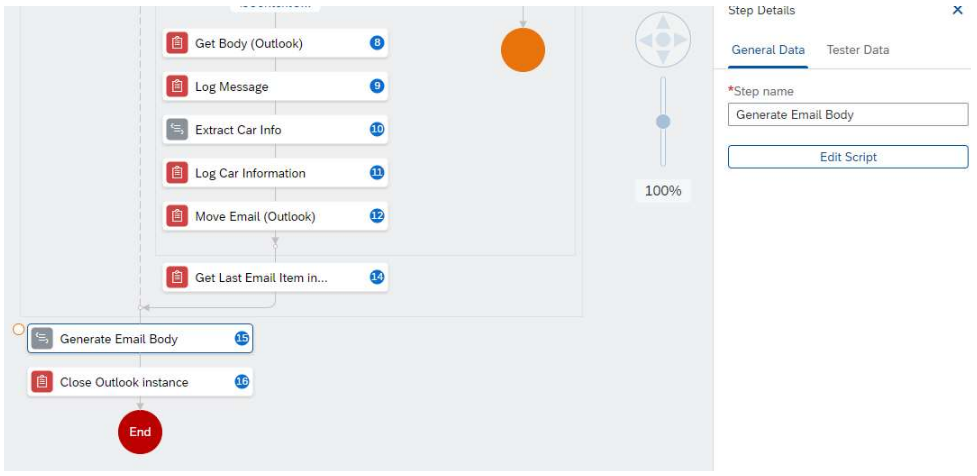
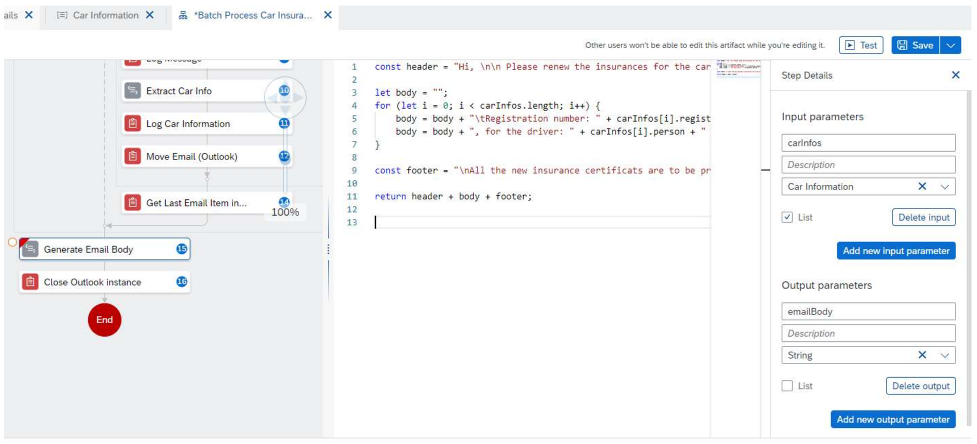
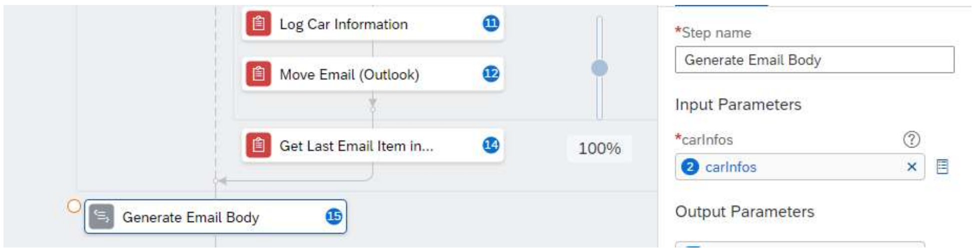
Drag and drop again the activity **Log Message** to add it as a step in the Automation, right after the step **Extract Car Info**.

Click on it and to rename it to **Log Car Information**. The type of Log is **Info**.

Click on the icon at the right of the **message** input to open the **expression editor**.



	 *Step name: Log Car Information Input Parameters *message: A-z type: Info label
Enter the expression `Step10.carInfo.person + "--" + Step10.carInfo.registrationNumber` And click on **Save expression** to save it and go back to the automation.	 Edit Expression `Step10.carInfo.person + "--" + Step10.carInfo.registrationNumber` Variables Operators Functions For more information, click here. Save Expression · Test · Cancel
Search the activity **Move Email (Outlook)** and drag and drop it right after the Step **Log Car Information**. Click on it to edit. The folder input is **Car Insurance renewal processed**. The folder type is **olFolderInbox**.	 Log Message 9 Extract Car Info 10 Log Car Information 11 Move Email (Outlook) 12 Get Last Email Item in... 14 *Step name: Move Email (Outlook) Input Parameters folder: Car Insurance renewal processed folderType: olFolderInbox storeName: A-z

Now we'll send an email with all collected car information to renew the insurance. Add a new **Custom Script** after the **Forever** block. Rename it to **Generate Email Body**. Click on **Edit Script** to add details.	
In the custom script editor, add an input parameter that named **carInfos**, the type is **list** of **Car Information**. Add an output parameter named **emailBody** of type **String**. In the code editor, copy-past the code located in the file ***[INT165] Custom Script – Generate Email Body.txt***	
Go back to the automation and click on the Step **Generate Email Body**. Bind the input with the output from **Step 2 – carInfos**.	

Step Details

General Data | Tester Data

*Step name: Generate Email Body

Edit Script

Other users won't be able to edit this artifact while you're editing it.

```
const header = "Hi, \n\n Please renew the insurances for the car
let body = "";
for (let i = 0; i < carInfos.length; i++) {
    body = body + "\tRegistration number: " + carInfos[i].regist
    body = body + ", for the driver: " + carInfos[i].person + "
}

const footer = "\nAll the new insurance certificats are to be pr

return header + body + footer;

```

Step Details

Input parameters

carInfos | Description | Car Information | ✓ List

Output parameters

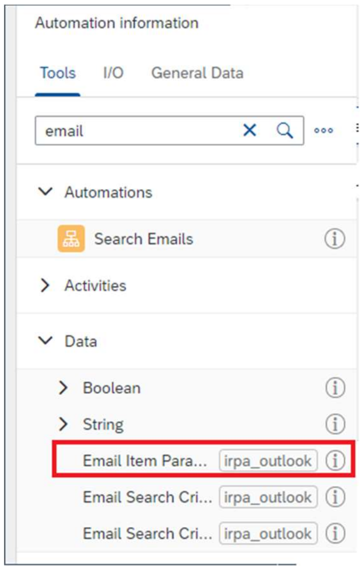
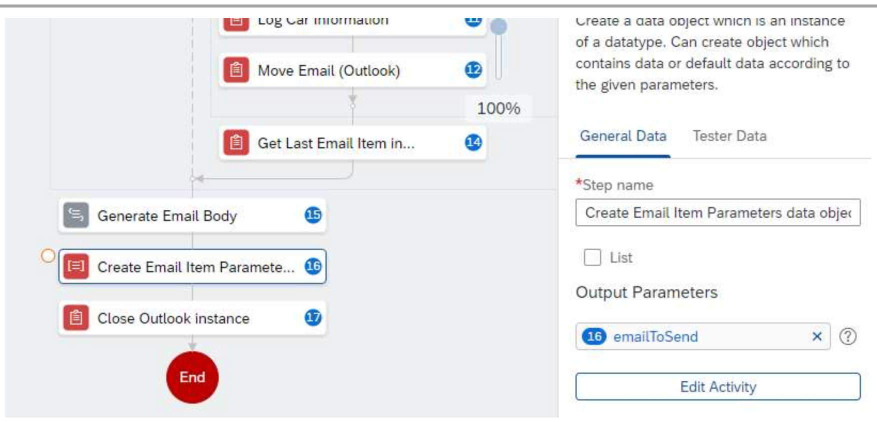
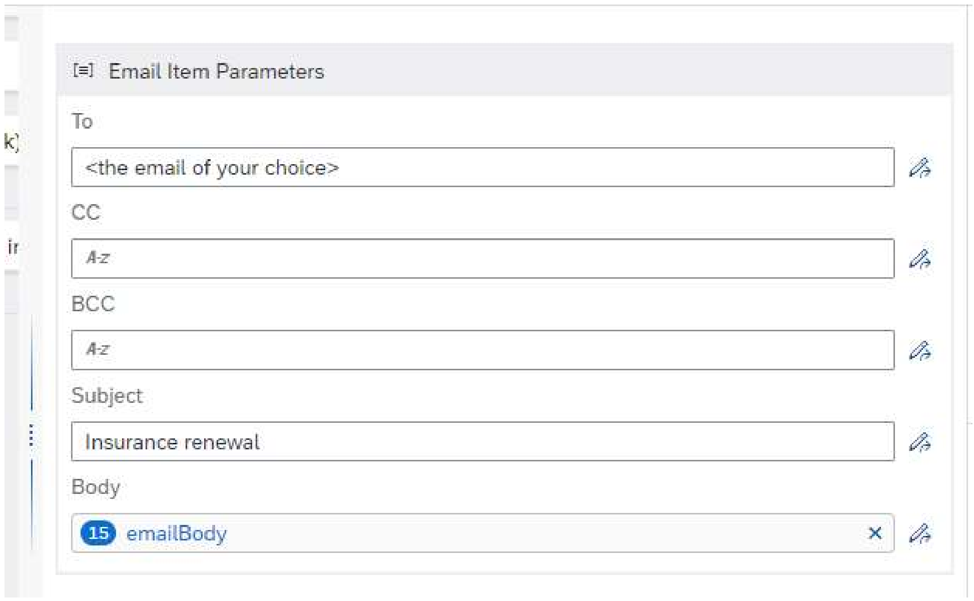
emailBody | Description | String | List

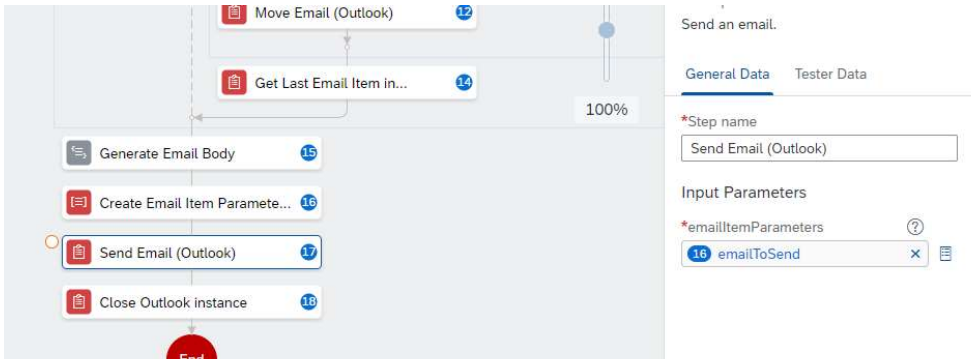
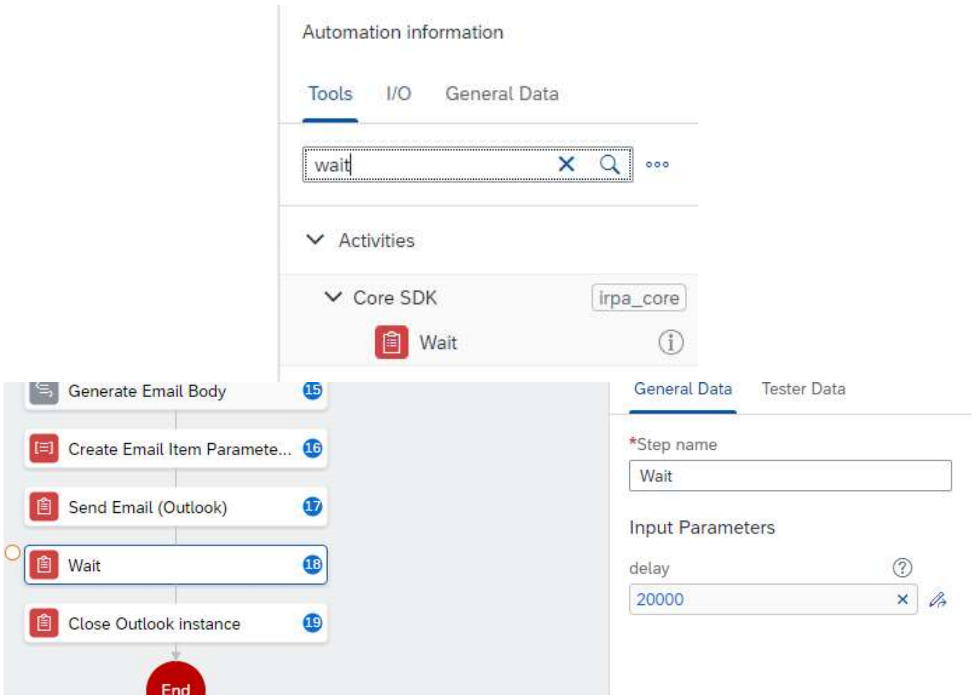
*Step name: Generate Email Body

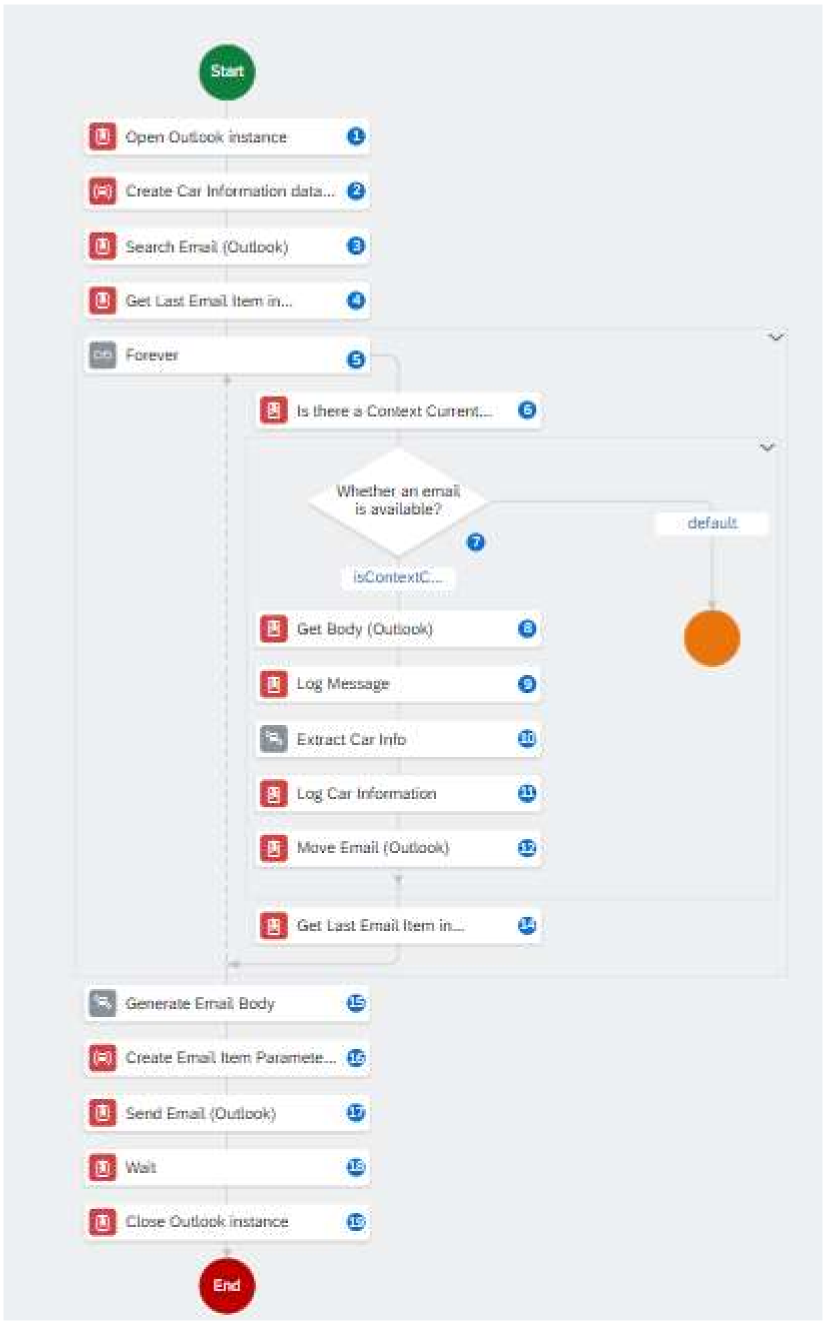
Input Parameters

*carInfos: 2 carInfos

Output Parameters

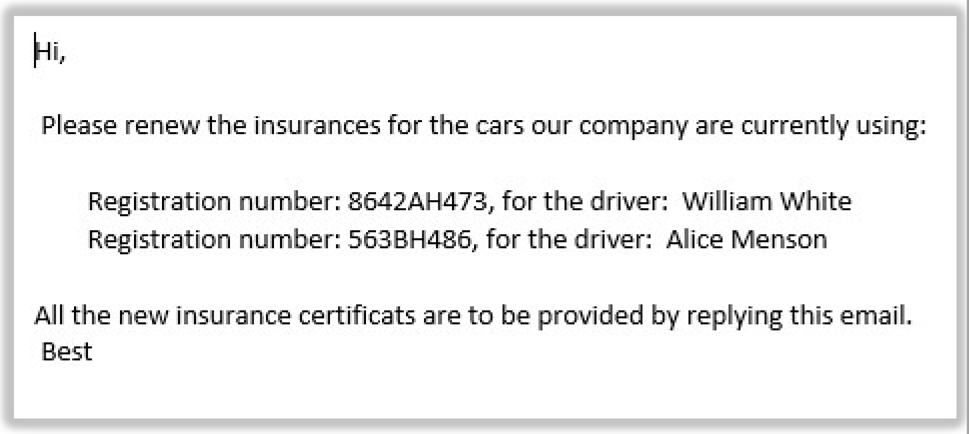
Localize now the data type **Email Item Parameters**. Drag and drop this data type in the automation right after the step **Generate Email Body**.	 Automation information Tools I/O General Data email Automations Search Emails Activities Data Boolean String Email Item Para... irpa_outlook Email Search Cri... irpa_outlook Email Search Cri... irpa_outlook
Click on the just added step **Create Email Item Parameters** and in the right panel, click on **Edit Activity** to fill the instance: • To: the email address of your choice that will receive the email • **Subject**: **Insurance renewal** • **Body**: output from **Step 15** Rename the output parameter to **emailToSend**.	 Log Car information 11 Move Email (Outlook) 12 100% Get Last Email Item in... 14 Generate Email Body 15 Create Email Item Paramete... 16 Close Outlook instance 17 End Create a data object which is an instance of a datatype. Can create object which contains data or default data according to the given parameters. General Data Tester Data *Step name Create Email Item Parameters data objec List Output Parameters 16 emailToSend Edit Activity  Email Item Parameters To <the email of your choice> CC A-z BCC A-z Subject Insurance renewal Body 15 emailBody

Search the activity **Send Email (Outlook)**. Drag and drop it to the automation right after the step **Create Email Item Parameters**. Edit the activity and put as input the output from Step 17 – **emailToSend**.	 Send an email. General Data / Tester Data. *Step name: Send Email (Outlook). Input Parameters: *emailItemParameters: 16 emailToSend
Search the activity **Wait**. Drag and drop it in the automation right after **Send Email (Outlook)**. Edit it and put **2000 ms** so to wait to the outlook to effectively sending the email.	 Automation information. Tools / I/O / General Data. Search: wait. Activities > Core SDK (irpa_core) > Wait. General Data / Tester Data. *Step name: Wait. Input Parameters: delay: 20000

At the end, your automation should look at this. **Save** it.	
Test the automation. As the result from the testing: • The individual insurance renew emails are moved to the folder	

Car Insurance renewal processed.

- An email that regroup all the requests is sent as shown by the screenshot.
- Some logs can be found in the Test console.

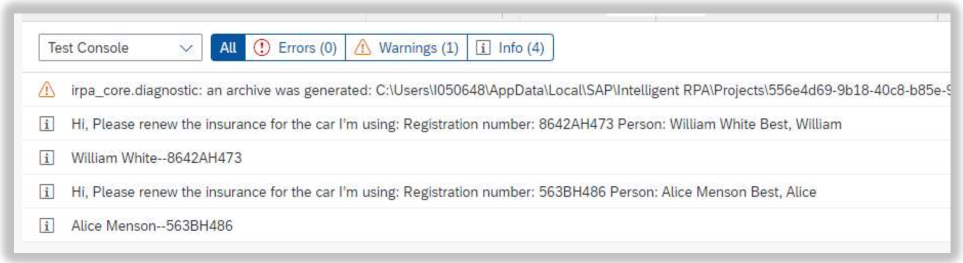


Hi,

Please renew the insurances for the cars our company are currently using:

Registration number: 8642AH473, for the driver: William White
Registration number: 563BH486, for the driver: Alice Menson

All the new insurance certificats are to be provided by replying this email.
Best



```
irpa_core.diagnostic: an archive was generated: C:\Users\I050648\AppData\Local\SAP\Intelligent RPA\Projects\556e4d69-9b18-40c8-b85e-
Hi, Please renew the insurance for the car I'm using: Registration number: 8642AH473 Person: William White Best, William
William White--8642AH473
Hi, Please renew the insurance for the car I'm using: Registration number: 563BH486 Person: Alice Menson Best, Alice
Alice Menson--563BH486
```

CONCLUSION

Congratulations! You have completed the exercise that allows to interact with MS Office Outlook in Intelligent RPA 2.0. In this exercise, you have learnt how to:

- Search and browse through emails in Outlook
- Move emails
- Prepare and send emails
- Create a data type in Intelligent RPA 2.0 and use it
- Use custom scripts

REFERENCE

Help portal for SAP Intelligent Robotic Process Automation: https://help.sap.com/viewer/p/IRPA
SAP Intelligent Robotic Process Automation on SAP Community:
https://community.sap.com/topics/intelligent-rpa

www.sap.com/contactsap

THE BEST RUN SAP®