

PUBLIC

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

INT165

Exercise 2: Integration of Word and Excel
Xiaohui XUE / SAP
Moumita Bera

TABLE OF CONTENTS

INTRODUCTION3

CREATION OF THE WORD AUTOMATION.....4

ADDING EXCEL ACTIVITIES..... 13

RUN THE BOT 18

SUMMARY21

REFERENCES21

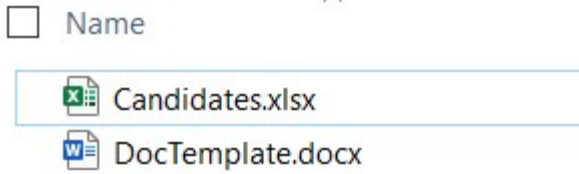
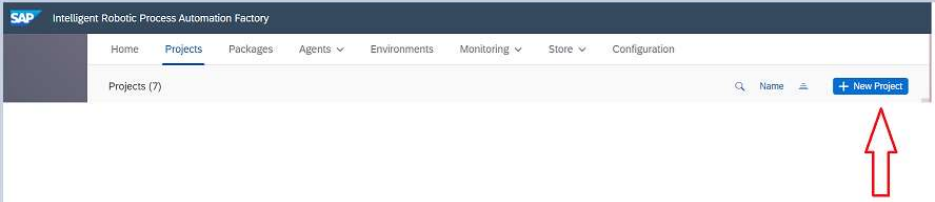
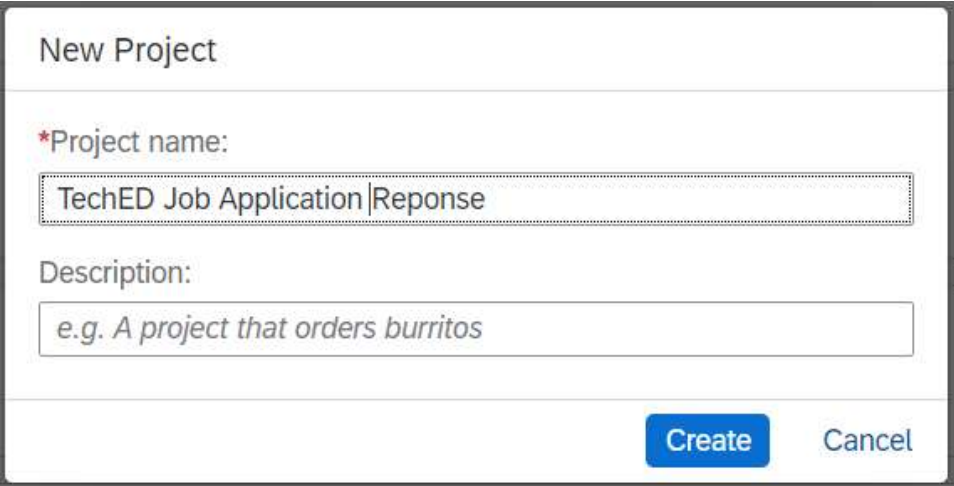
INTRODUCTION

This workshop aims to introduce the new features available in SAP Intelligent RPA 2.0 and how they can be used to simplify the business integration with Microsoft office and with SAP tools.

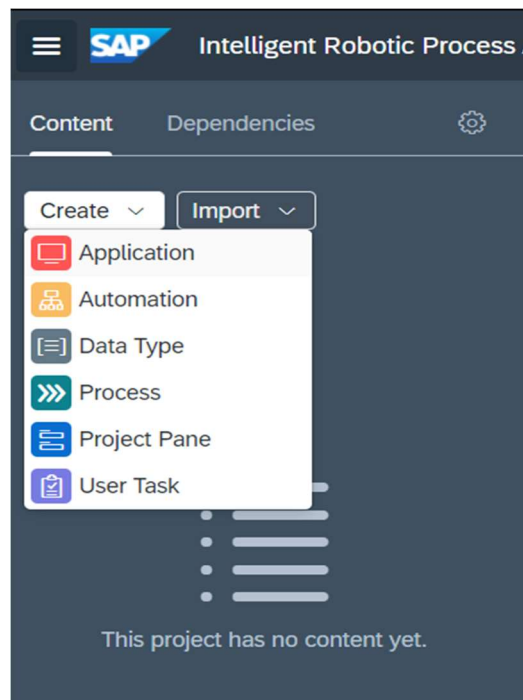
This exercise shows how to create an automatic process to handle job applicants. Given the list of candidates in an excel, the workflow automatically creates replies using a WORD document as template.

Estimated time: 30 minutes

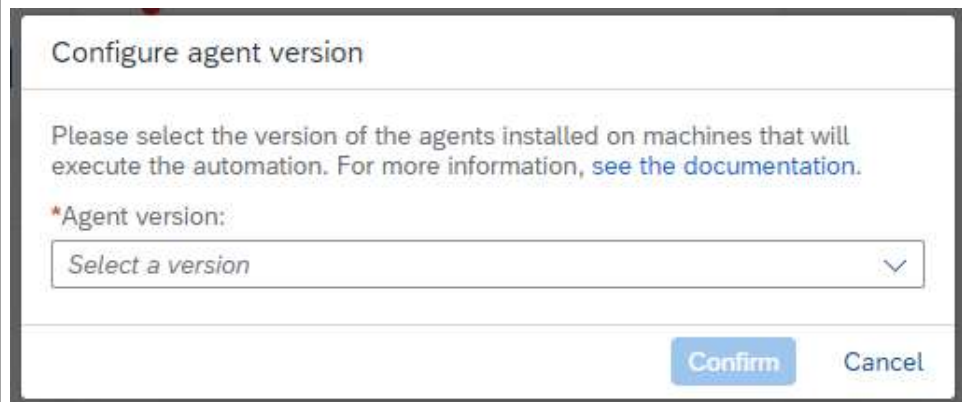
CREATION OF THE WORD AUTOMATION

Explanation	Screenshot
<p>Download the Excel and WORD file provided with this exercise and save it in a local folder. "C:\test". [If "test" folder is not in system then please create it]</p>	
<p>Go to Projects Tab. Click on New Project</p>	
<p>A new popup appears. Choose a name for the project (here TechEd Job Application Response), then click on the Create button.</p>	

In the project, **click** on the Create button to expand the dropdown menu, then **select** Automation.



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**.



A new popup appears.
Choose a name for the automation (here Document Generation), then **click** on the Create button.

This will create the automation and open the automation editor.

Create Automation

***Name:**

***Identifier:**

Description:

Triggerable:
☒

[Create](#) [Cancel](#)

Click on Project Settings icon and select the section **Dependencies**.

Two dependencies should be automatically added which are irpa_core and irpa_excel.

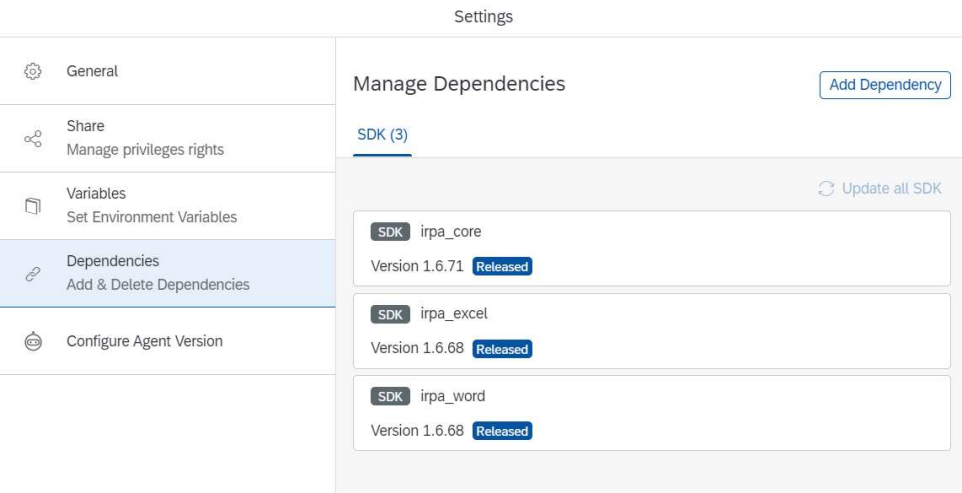
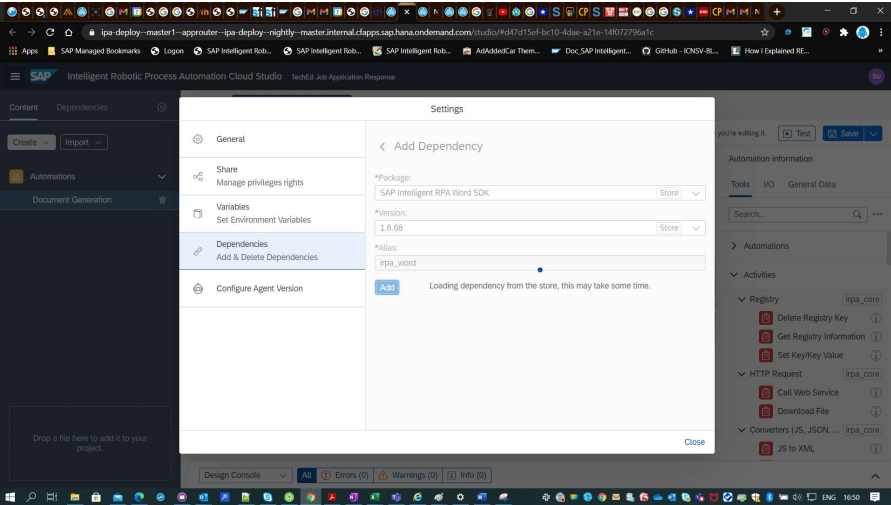
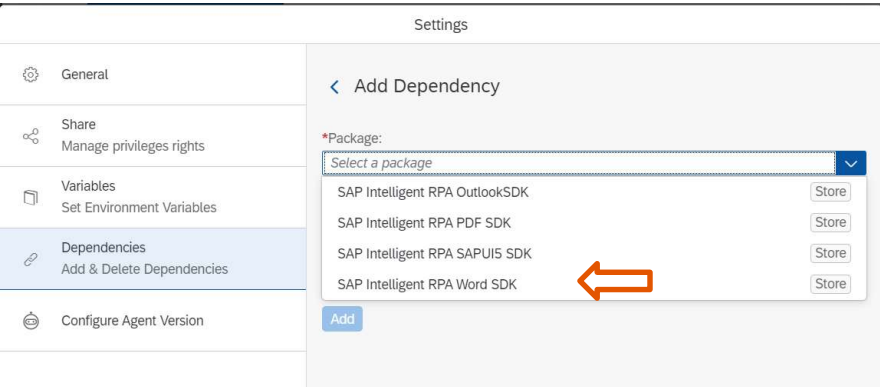
Click on **Add Dependency** icon.

The screenshot shows the SAP Intelligent Robotic Process Automation Cloud Studio interface. The top bar includes the SAP logo and the text "Intelligent Robotic Process Automation Cloud Studio" and "TechED outlook". The left sidebar contains the "Project Explorer" with sections for "Content" and "Dependencies". The "Dependencies" section is active, showing a list of dependencies: "irpa_core" (Version 1.6.71, Released) and "irpa_excel" (Version 1.6.68, Released). The "Settings" panel is open on the right, showing the "Manage Dependencies" section with an "Add Dependency" button. The "Dependencies" section is selected in the left sidebar, and the "Add & Delete Dependencies" option is highlighted.

From Package dropdown select SAP Intelligent RPA Word SDK. Select available Version.

Click on Add button. This will add Word library in your current project scope.

Finally, click on close.



In the right panel, tab I/O, click on Add new input Parameter

Automation information

Tools **I/O** General Data

Input parameters

No input parameter defined for this Automation.

Add new input parameter

Output parameters

No output parameter defined for this Automation.

Add new output parameter

Create an input parameter
Name of String type.

you're editing it.

Test

Save



Automation information

Tools **I/O** General Data

Input parameters

Name

Description

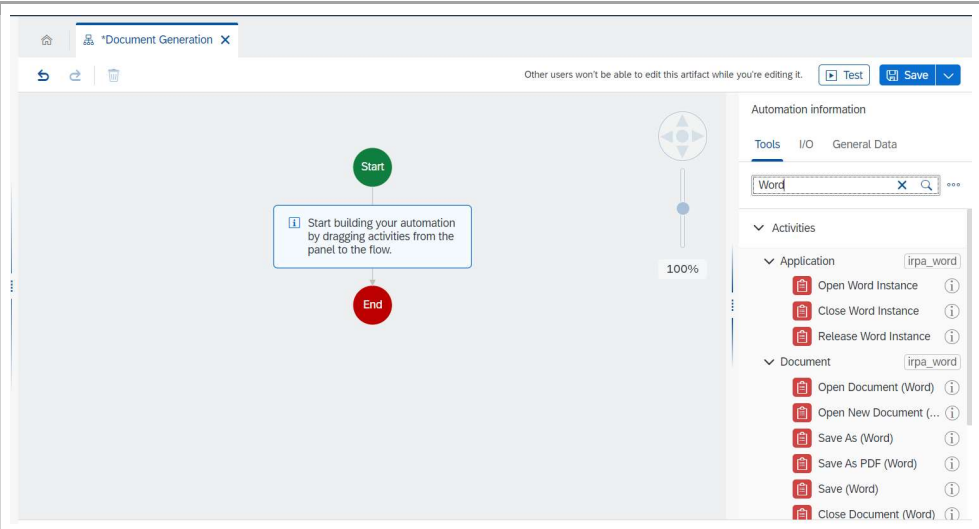
String



☐ List

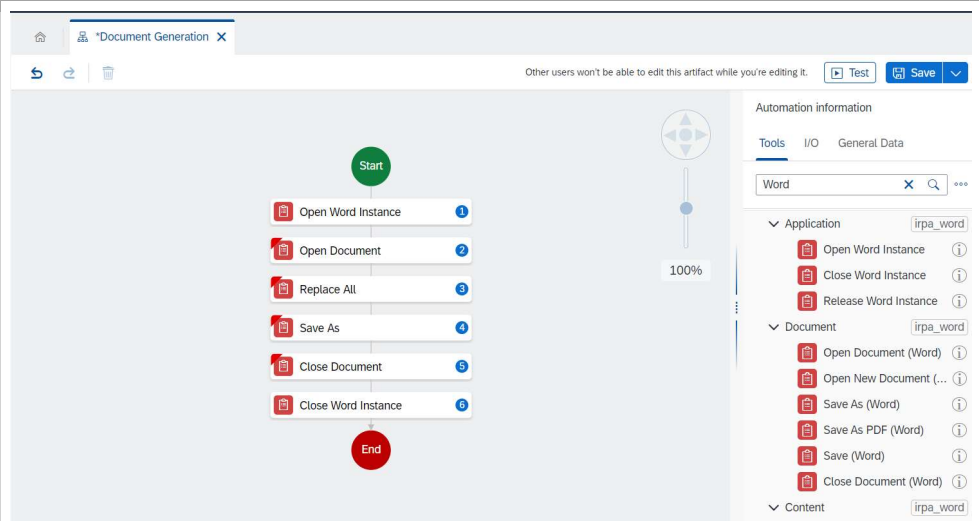
Delete input

In the right panel of the Automation editor, **filter** the activities with the keyword **Word** to find related activities.

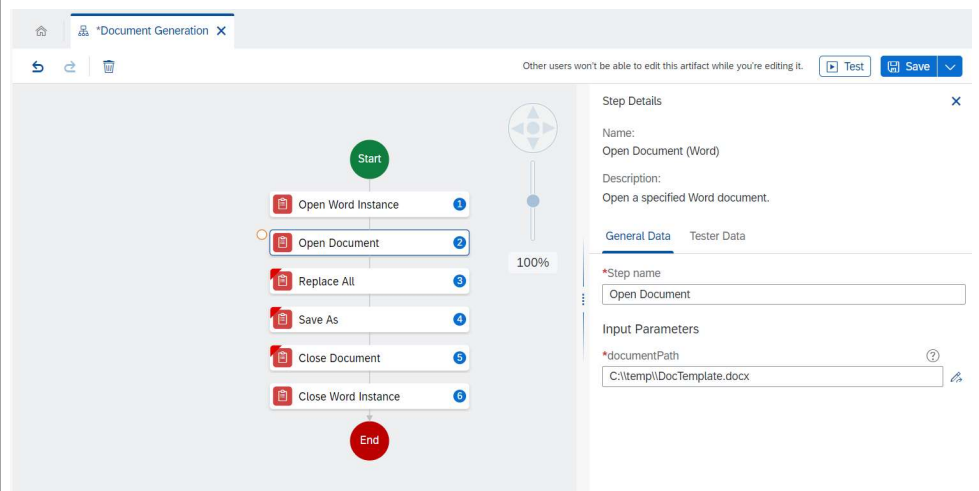


Drag and drop these activities in the canvas:

- Open Word instance
- Open Document
- Replace All
- Save As
- Close Document
- Close Word instance



Click on **Open Document** activity, in the right panel, enter the parameter which is the local path of the word document given with the exercise.



Click on Replace All activity.

The parameter **oldString**'s value should be **[NAME_PH]**. This is the placeholder defined in the template WORD document.

The parameter **newString** should take the input parameter **Name**.

Also configure the parameter **advancedSearchParameters** and enter **wrap = 1**.

The first screenshot shows the 'Document Generation' workflow with steps: Start, Open Word Instance, Open Document, Replace All, Save As, Close Document, Close Word Instance, and End. The 'Replace All' step is selected, and the 'Step Details' panel on the right shows the 'Input Parameters' section. The 'oldString' parameter is set to '[NAME_PH]'.

The second screenshot shows the same workflow, but the 'newString' parameter is now set to 'Name'.

The third screenshot shows the 'Edit data' panel for the 'advancedSearchParameters' section. The 'wrap' parameter is set to '1'.

Click on **Save As** activity.

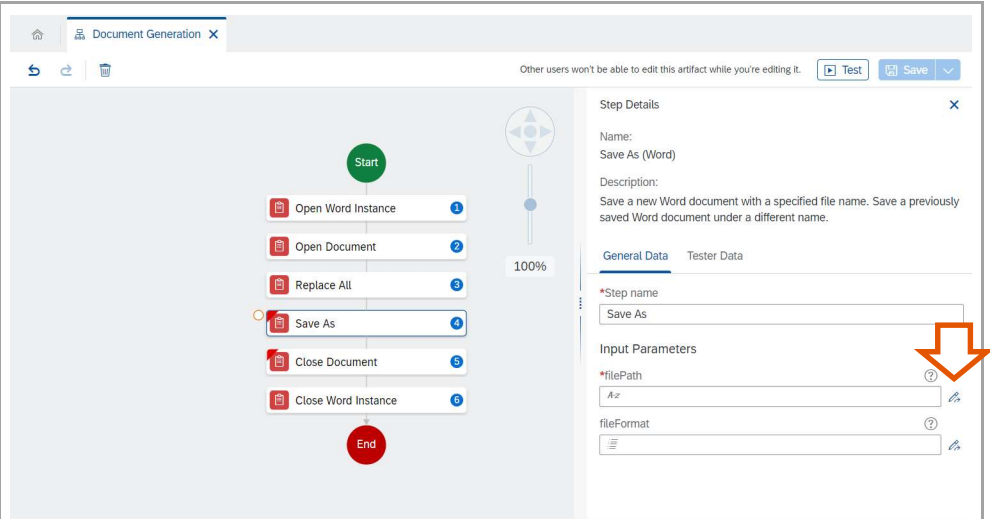
For the parameter **filePath**, click on the **expression editor** icon at the right.

It will open a popup

Put a path as you have decided, using the input parameter **Name**.

Test and Save Expression.

For the **fileFormat** parameter, choose **wdFormatPDF** value in the dropdown list.



Document Generation

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

Step Details

Name: Save As (Word)

Description: Save a new Word document with a specified file name. Save a previously saved Word document under a different name.

General Data Tester Data

*Step name: Save As

Input Parameters

*filePath: [Expression Editor Icon] [?] [x]

fileFormat: [?] [x]

Workflow steps: Start, Open Word Instance, Open Document, Replace All, Save As, Close Document, Close Word Instance, End.

Edit Expression

'c:\\temp\\' + Step0.Name + '.pdf'

> Variables
> Operators
> Functions

For more information, [click here.](#) [Save Expression] [Test] [Cancel]

Document Generation

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

Step Details

Name: Save As (Word)

Description: Save a new Word document with a specified file name. Save a previously saved Word document under a different name.

General Data Tester Data

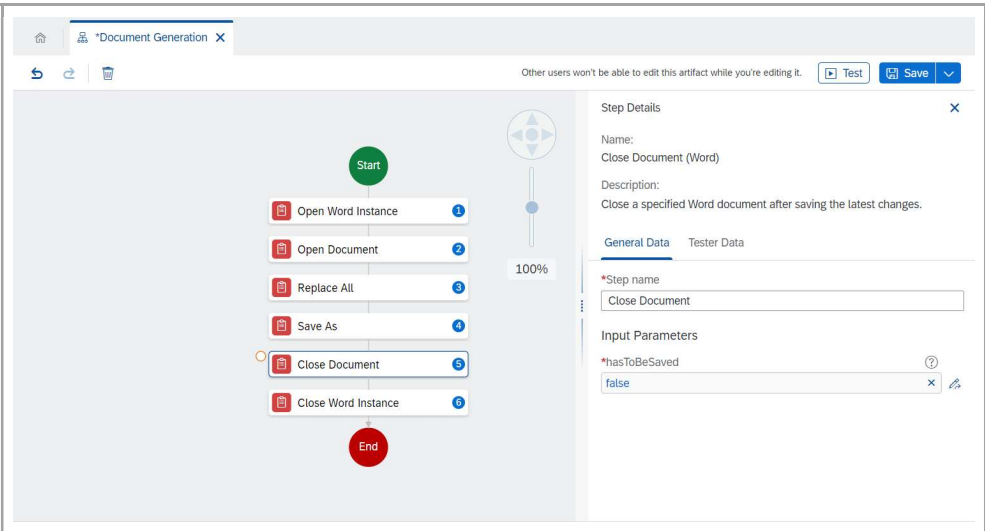
*Step name: Save As

Input Parameters

*filePath: "C:\\temp\\" + Step0.Name + ".pdf" [x]

fileFormat: wdFormatPDF [x]

Workflow steps: Start, Open Word Instance, Open Document, Replace All, Save As, Close Document, Close Word Instance, End.

<p>Click on Close Document activity.</p> <p>Enter the value of hasToBeSaved parameter as false.</p>	 <p>The screenshot shows a workflow editor for 'Document Generation'. The workflow steps are: Start, Open Word Instance, Open Document, Replace All, Save As, Close Document, Close Word Instance, and End. The 'Close Document' step is highlighted. On the right, the 'Step Details' panel shows the step name 'Close Document' and the input parameter 'hasToBeSaved' set to 'false'.</p>
<p>This automation is complete, you can save and test it.</p> <p>It should create a PDF for each execution. The content of the execution is the initialization of the template using the input name.</p>	

ADDING EXCEL ACTIVITIES

Create another automation called Batch Document Generation.

Create Automation

*Name:

Batch Document Generation

*Identifier:

batchDocumentGeneration

Description:

e.g. An automation that books flights

Triggerable:

Create

Cancel

In this Automation editor, add two activities:

- Open Excel instance
- Open Workbook

For the **Open Workbook** activity, enter the value of parameter **workbookPath** with the path of the excel file provided with the exercise.

SAP

Intelligent Robotic Process Automation Cloud Studio

TechEd Job Application Response

Content

Dependencies

Create

Import

Automations

Batch Document Generation

Document Generation

Start

Open Excel Instance

Open Workbook

End

100%

Step Details

Name:

Open Workbook

Description:

Open a workbook referenced by a workbook path. Once opened, use the Activate Workbook activity for further usage.

General Data

Tester Data

*Step name

Open Workbook

Input Parameters

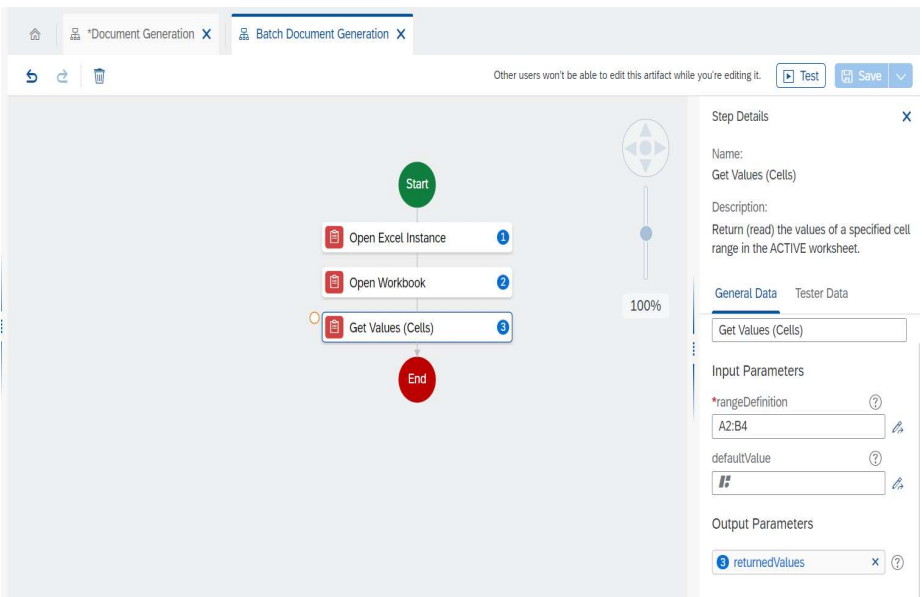
*workbookPath

c:\templ\Candidates.xlsx

Add a new Get Values (Cells) activity.

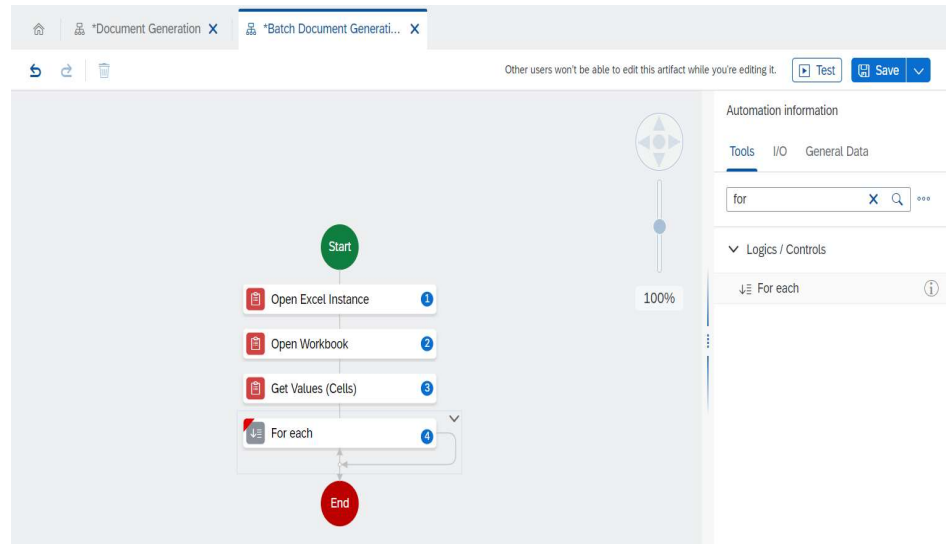
For the parameter **rangeDefinition**, enter the value **A2:B4**.

It corresponds to the cells that provides the job application candidates.



Drag and drop the **For each** logics/control activity.

Set the looping list as **returnedValue** from the Step 3. This corresponds to the list of candidates retrieved from the Excel file.



Now drag and drop the first automation, namely Document Generation in the loop from the Right hand side Panel.

The screenshot shows an automation editor interface. The main workspace contains a workflow diagram with the following steps: a green 'Start' circle, followed by three red document icons labeled 'Open Excel Instance' (1), 'Open Workbook' (2), and 'Get Values (Cells)' (3). Below these is a 'For each' loop (4), which is currently empty. The loop ends with a red 'End' circle. On the right side, there is a panel titled 'For each' with a close button (X). Under 'General Data', the 'Step name' is 'For each'. Under '*Set looping list', the value is 'returnedValues'. Under 'Loop Parameters', there are fields for 'currentMember' and 'index'. At the bottom, there is an 'Output parameter' section with an 'Add Parameter' button. A zoom slider is visible on the right side of the workspace, set to 100%.

The screenshot shows the same automation editor interface, but now the 'For each' loop (4) contains a 'Document Generation' step (5), represented by a red document icon. The right panel is now titled 'Automation information' and has tabs for 'Tools', 'I/O', and 'General Data'. Under the 'Automations' section, there is a search bar and a list of automations: 'Batch Document Generation' and 'Document Generation'. Below this, there are expandable sections for 'Activities', 'Data', and 'Logics / Controls'. An orange arrow points to the 'Document Generation' automation in the list.

Use the **expression editor** to provide the input value for the automation **Document Generation**.

This corresponds to the concatenation of first name and last name for each candidate.

The expression editor provides auto completion to help putting the accurate statement.

The screenshot displays the UiPath Studio interface. On the left, a workflow diagram shows a sequence of steps: 'Start' (green circle), 'Open Excel Instance' (1), 'Open Workbook' (2), 'Get Values (Cells)' (3), 'For each' (4), 'Document Generation' (5), and 'End' (red circle). The 'For each' loop is expanded, showing the 'Document Generation' step. On the right, the 'Step Details' panel is open for the 'Document Generation' step. It shows the 'Name' as 'Document Generation' and the 'Description' as 'No value'. The 'General Data' tab is selected, showing the '*Step name' as 'Document Generation'. The 'Input Parameters' section shows the '*Name' parameter with the expression 'Step4.currentMember + ' ' + Step4.currentMember[1]'. Below the workflow, the 'Edit Expression' dialog is open, showing the same expression: 'Step4.currentMember[0] + ' ' + Step4.currentMember[1]'. The dialog has a search bar and a list of categories: 'Variables', 'Operators', and 'Functions'. At the bottom of the dialog, there is a link 'For more information, click here.' and buttons for 'Save Expression', 'Test', and 'Cancel'.

Step Details

Name:
Document Generation

Description:
No value

General Data Tester Data

*Step name
Document Generation

☐ Trigger as a new job

Input Parameters

*Name
Step4.currentMember + ' ' + Step4.currentMember[1]

Edit Expression

Step4.currentMember[0] + ' ' + Step4.currentMember[1]

> Variables
> Operators
> Functions

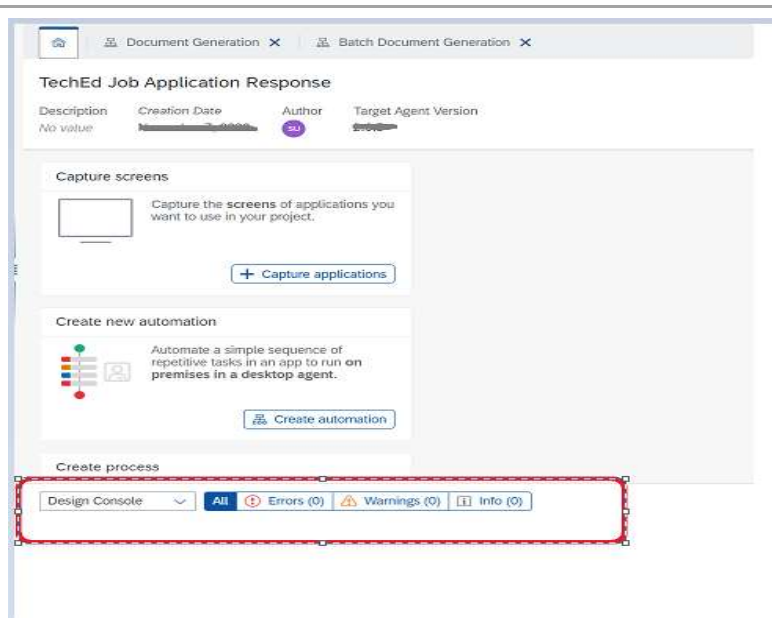
For more information, [click here](#).

Save Expression Test Cancel

<p>After the loop, add two other activities to close the Excel (namely, close Workbook and Close Excel Instance).</p> <p>For the Close Workbook activity, the parameter hasToSave should be false.</p>	<div><div><div><div><div>Start</div><div>Open Excel Instance 1</div><div>Open Workbook 2</div><div>Get Values (Cells) 3</div><div>For each 4</div><div>Document Generation 5</div><div>Close Workbook 6</div><div>Close Excel Instance 7</div><div>End</div></div><div>100%</div></div><div><div>Step Details</div><div><div>Name: Document Generation</div><div>Description: No value</div><div>General Data Tester Data</div><div>*Step name Document Generation</div><div><input type="checkbox"/> Trigger as a new job</div><div>Input Parameters</div><div>*Name Step4.currentMember + ' ' + Step4.currentMember[1]</div></div></div></div></div>
<p>You can now save and test the automation Batch Document Generation.</p> <p>It should generate as much responses as listed in the excel file.</p>	<div><div>Other users won't be able to edit this artifact while you're editing it. <div>Test Save</div></div><div>Other users won't be able to edit this artifact while you're editing it. <div>Test Save</div></div><div>Test Automation<div>*Environment: Select an environment</div></div><div><div>Test Cancel</div><div><div>Alice Norton.pdf</div><div>Candidates.xlsx</div><div>DocTemplate.docx</div><div>John Barker.pdf</div><div>Paul Smith.pdf</div></div></div></div>

RUN THE BOT

- Go to your project.
- Click on the **Home** button of the project
- Check that the conditions to generate a Package are fulfilled (ie. All values for error and warning are set to 0)



- Click on **Generate Package** button on the right side.
- A dialog will open. Click again on **Generate Package**. Close the Project navigator tab

Generate Package

***Name:**

Description:

Version Number:

Locate the generated package in the **Packages** tab.

Click on **Add Trigger** to create a scheduled trigger:

- In “Select an Environment”, select the Environment you have previously created. Click on “Next”
- In “Select a trigger type” step, chose **Scheduled** trigger

HomeProjectsPackagesAgents ▾Environme

Packages (1)

TechED Excel Demo from

Version 1.0.0 BETA

Preview

Create Trigger

1 Select an environment. — 2 Deploy — 3 Set Environment Variables — 4 Select a trigger type

TechEd

TechEd Demo Environment

Test

Author

su

Updated just now.

Create Trigger

1 Select an environment. — 2 Deploy — 3 Set Environment Variables — 4 Select a trigger type

RPACHallenge1 Version 1.0.0

API

A trigger of type API opens a dedicated endpoint that allows an external application to execute a scenario or a process.

Attended

With an attended trigger, the deployed package is distributed to a specific group of agents, and users run the jobs manually.

Scheduled

With a scheduled trigger, the user does not need to start a job manually. Jobs are created according to the schedule you define in the trigger.

Previous

Next

Cancel

19

- For Execute field, select from dropdown your automation to execute. Here: "Batch Document Generation"
- Select **Date Range** from yesterday until tomorrow and some Time period including today
- Click **Create**

Add Scheduled Trigger

*Name: Scheduled_01

Description: e.g. A scheduled trigger

*Execute: Batch Document Generation

*Priority: Automation

*Date range: Nov 7, 2020 to Nov 14, 2020

*Timezone: (UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi

*Recurrence: 5 Minutes

*Jobs expire after: 60 Minutes

New job: ☒ A new job will be added no matter what happens.

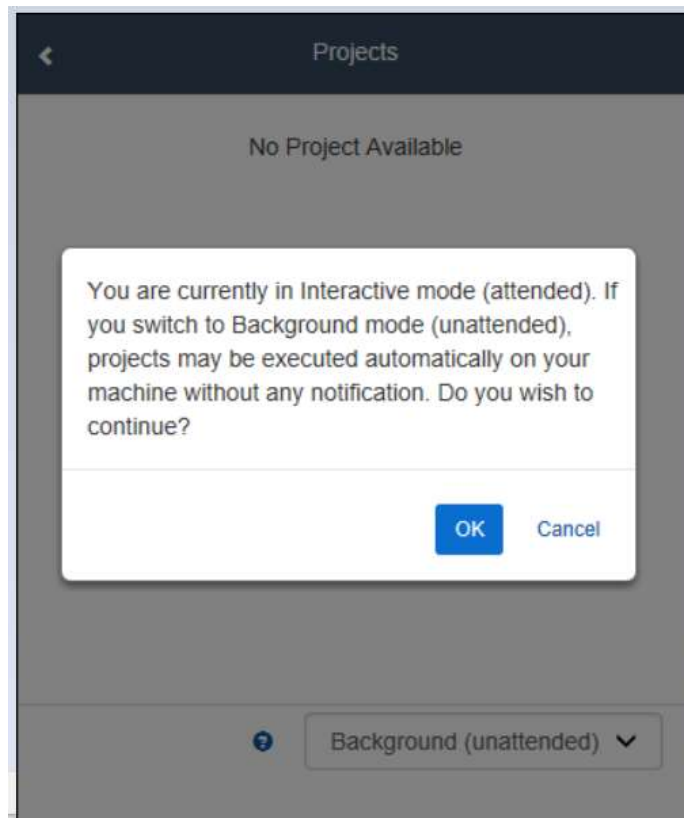
Midnight (AM) Noon (PM)

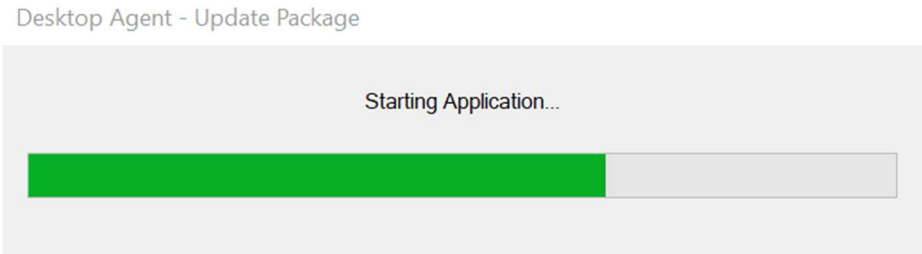
Sunday

Monday

Create Cancel

- Start Desktop Agent (You will see a popup which indicates that the package is downloading)
- From the **Factory**, check in Agents/Agents table, the default mode is Attended by default.
- From the **systray**, click on projects,



then select 'Background (unattended)' and click on OK.	
Execute unattended process Wait until job gets distributed and scenario gets executed successfully (3 – 5 min)	 <p>The screenshot shows a window titled 'Desktop Agent - Update Package'. Inside, there is a light gray box with the text 'Starting Application...' and a green progress bar that is approximately 75% full.</p>

SUMMARY

You have completed the exercise!

You are now able to:

- ... create an automation using the Cloud Studio
- ... use automations to manipulate Excel sheets
- ... create an automation to manipulate word documents
- ... test and run your project

REFERENCES

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

© 2020 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See www.sap.com/copyinginfo for additional trademark information and notices.