

IBM Cloud Pak for Business Automation Demos and Labs 2022

IBM RPA and Workflow Integration

V 2.2

Bu Feng Hou

houbf@cn.ibm.com

Paul Pacholski

pacholsk@ca.ibm.com

Olaf Hahnl

olaf.hahnl@de.ibm.com

NOTICES

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

TRADEMARKS

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates. Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in

the U.S. and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

© Copyright International Business Machines Corporation 2020.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Table of Contents

1	INTRODUCTION.....	5
2	OVERVIEW.....	5
2.1	PRE-REQUISITES	6
2.2	REFERENCES.....	7
3	ACCESSING THE ENVIRONMENT	8
3.1	RESERVE ENVIRONMENT.....	8
4	BUILD IT YOURSELF – STEP-BY-STEP INSTRUCTIONS	13
4.1	CREATE THE RPA ORCHESTRATION PROCESS.....	13
4.2	CREATE PROCESS TO INVOK RPA BOT ASYNCHRONOUSLY.....	15
4.2.1	<i>Explore the IBM RPA Toolkit</i>	20
4.2.2	<i>Develop a Workflow Process to start an RPA Bot.....</i>	26
4.2.3	<i>Verification Instructions</i>	40

1 Introduction

IBM Robotic Process Automation (RPA) provides a comprehensive set of Robotic Process Automation (RPA) features:

- **Unattended bots**
Use an RPA-driven digital workforce to automate repetitive tasks without human intervention.
- **Attended bots**
Remote Desktop Automation (RDA) enables a human workforce to augment work using bots to perform repetitive tasks on demand.
- **Intelligent Virtual Agent (IVAs) chatbots**
Combine chat and RPA commands to create chatbots through multiple channels that can provide engaging client interactions.
- **Optical Character Recognition (OCR)**
Process documents by extracting structured data from unstructured content.
- **Dashboards**
Gain business insights into business operations.

With IBM RPA, IBM can provide customers with additional benefits:

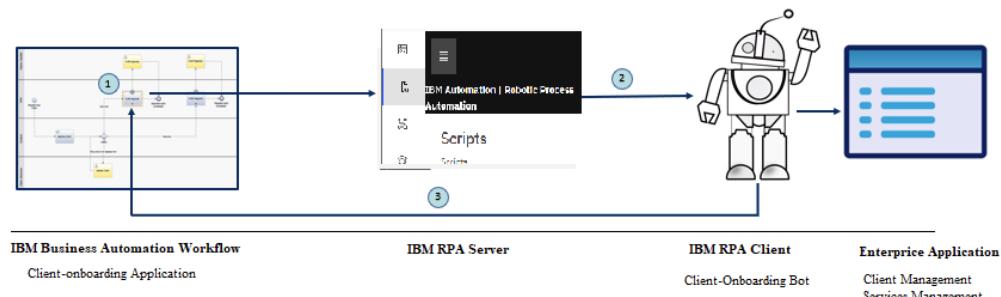
- **Faster time to value**
Speed and simplicity of purchasing and deploying through easier licensing.
- **A comprehensive platform to automate all types of use cases**
Tighter integrations between RPA and the rest of our platform.
- **Automate business and IT processes**
Expand our automation mission to IT use cases.
- **Operationalize AI**
Fulfill IBM's vision of operationalizing AI in every aspect of the business.

You can explore the [Documentation](#) to understand more details about IBM RPA.

2 Overview

In this lab, you will learn how to invoke a bot--authored in RPA Studio--from a process developed with Business Automation Workflow.

The integration steps between Workflow and RPA are as follows:



1. Workflow reaches the activity where a bot should be invoked. The process designer models the activity to call a service flow that uses an external service that invokes an

RPA bot using the IBM RPA REST APIs. IBM RPA offers two types of REST APIs to start bots. One is a synchronous unprotected REST API to run the bot via a client-side API. This can be used when the bot finishes its execution very quickly. Another approach is to start the bot in an orchestrator process via server-side asynchronous APIs in a protected way. In this lab, we will use the 2nd approach to start the bot from a workflow.

2. The RPA server deploys the bot by passing business data to the RPA Client.
3. The RPA client agent performs the actual work by executing the bot script. The bot script will add client-onboarding information and signed services into backend applications as part of the end-to-end client onboarding solution. Once the execution is completed, it will pass the output data back to the Workflow process with a status code indicating if the bot script execution was successful or not.

2.1 Pre-requisites

For this lab, you need to access:

- **IBM Robotic Process Automation:** You need to reserve a lab environment from IBM Technology Zone, as explained below.
- **IBM Business Automation Studio:** If you are performing this lab as a part of an IBM event (such as Technical Jam), access the document that lists the available systems and URLs along with login instructions. Or, you may not be able to perform this lab unless you have deployed your own IBM Cloud Pak for the Business Automation environment and imported the Client Onboarding Application.

All the pre-requisites have been pre-installed/configured in the lab template. The information below is just for information purposes.

IBM Products:

- IBM Robotic Process Automation v21.0.x.
- IBM Cloud Pak for Business Automation v21.0.x.

Custom Solutions/Code:

- The Client Onboarding Toolkit - contains the predefined business object definition.
- The IBM RPA Toolkit - contains the service flow to start the bot matching the information required by the two backend systems in an orchestrator process via server-side async APIs.

This toolkit is just a sample implementation of invoking the RPA server-side APIs. It simplifies the workflow and RPA integration for this lab only. You may want to study how it is implemented, but please don't directly use it in any real customer project.

- A Java swing application simulating the backend, third-party system for the Client Management System.

- A web application simulating the backend, third-party Services Management System for managing the services a client has signed up to.

2.2 References

1. [IBM Robotic Process Automation Documentation](#)
2. [IBM Robotic Process Automation Command Documentation](#)

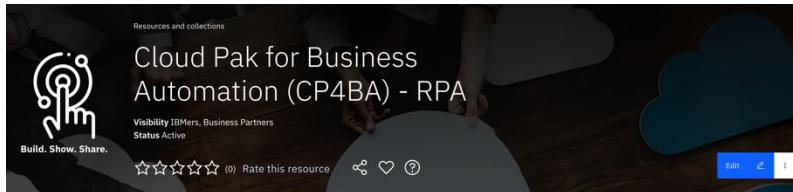
3 Accessing the Environment

If you have already reserved a lab environment from IBM Technology Zone, please go to [Chapter 4](#) directly.

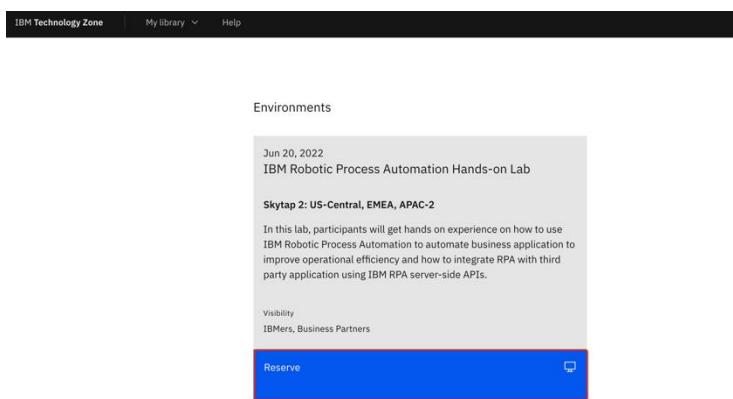
3.1 Reserve Environment

To get started with this lab, please follow below steps to reserve an environment:

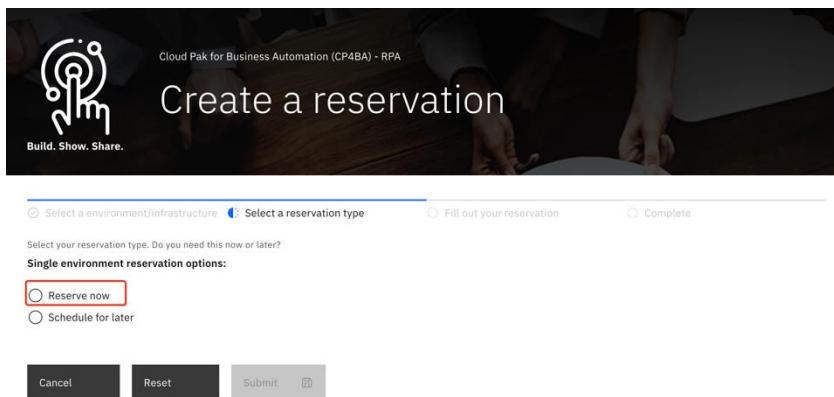
1. Click [here](#) to open IBM Technology Zone Reservation portal. You need to use your IBMID to login to the portal.



2. Click **Environments** on the left panel, then click the blue button.



3. Select **Reserve for now**, then click **Submit**.



4. On the reservation page, make the appropriate selections as below. Once done, click **Submit**.

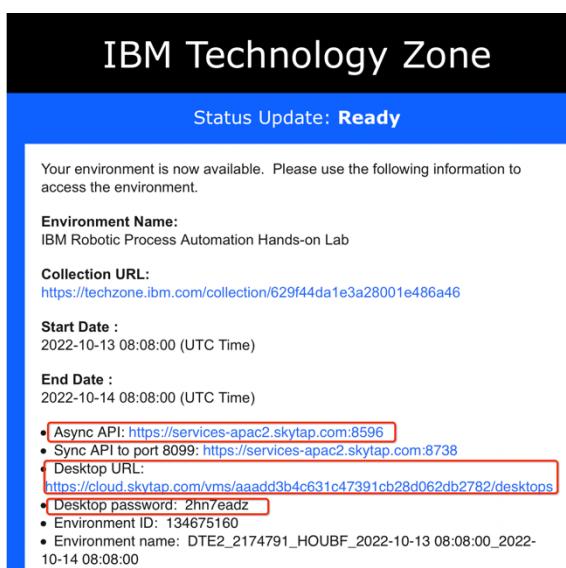
Purpose: Select Practice/Self-Education.

Purpose description: Enter something like **Self Education**.

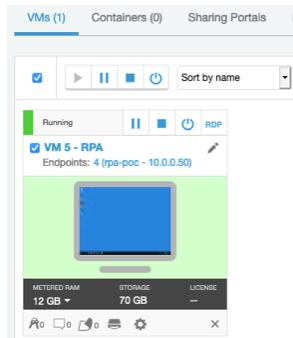
End date and time: Select the end date and time that the environment will be deleted.

Preferred Geography: Select the geography where your environment will be created. In order to get better network connection, suggest you select the same geography as where you locate in.

5. Once you have reserved an environment, you will receive an email with a link to access the management console for the environment including a password (**Desktop URL** and **password**). It also contains a URL to access the IBM RPA REST Service remotely(first URL circled in red as shown in the screenshot below).



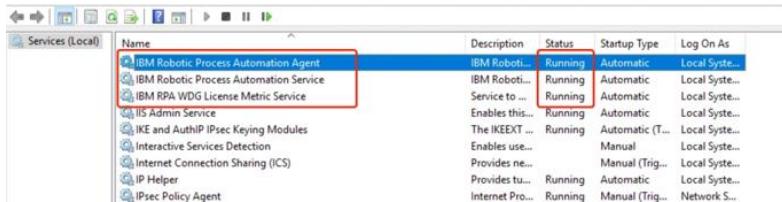
6. Click the desktop access link above to open your environment. When you are prompted to enter the environment password, please enter the desktop password above. Wait a few minutes, your environment will be started as below.



7. Click **VM 5 – RPA** to open the Windows environment in web browser.
8. Click Service icon  from Windows toolbar.



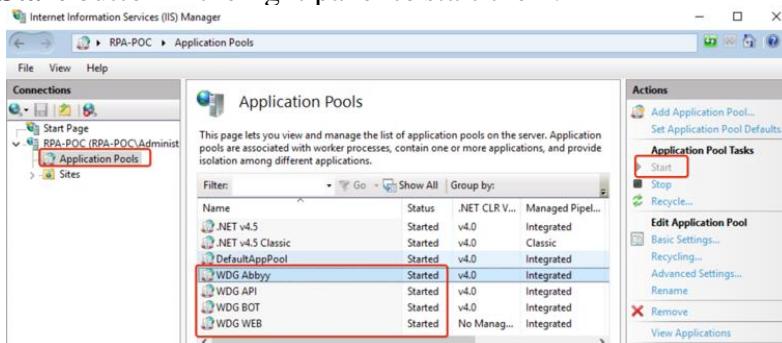
9. Check that the **IBM Robotic Process Automation Agent**, **IBM Robotic Process Automation Service**, **IBM RPA WDG License Metric Service** are in running status through Windows Service Manager. Those in stopped status please start.



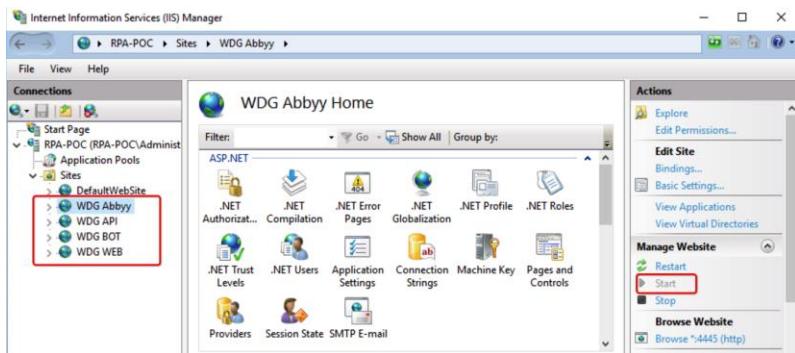
10. Click the Internet Information Service (IIS) manager icon  in the Windows toolbar.



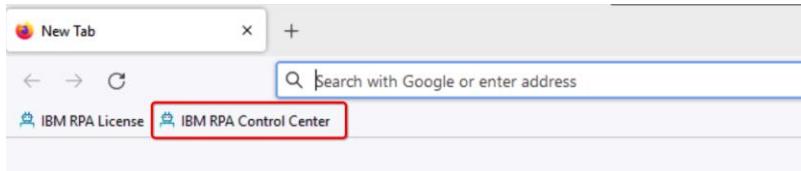
11. Check **application pools** are in started status. For those that are not started, click the **Start** button in the right panel to start them.



12. Check below **Sites** have been started. For those that are not started, click **Start** button in the right panel to start them.



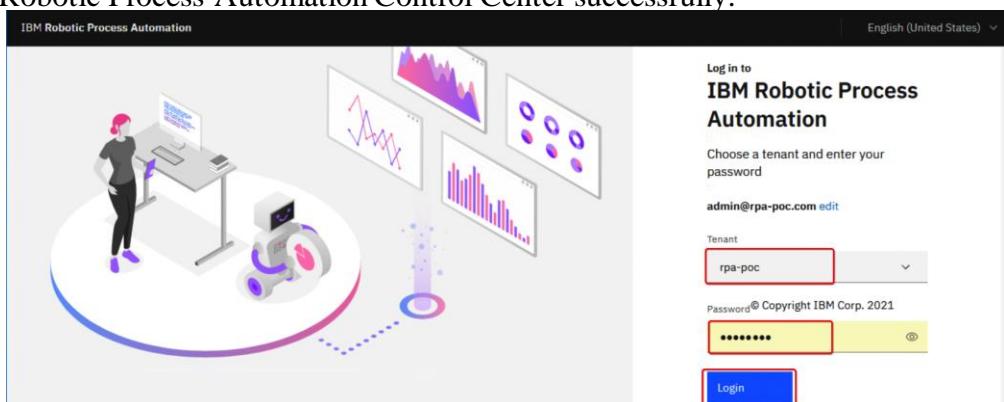
13. Start **Firefox** and click **IBM RPA Control Center** in the Firefox bookmark toolbar.



14. Enter [admin@rpa-poc.com](#) as user name and click the **Continue**.



15. Select **rpa-poc** as tenant, enter **passw0rd** (make sure to use a zero as part of the password) as password, then click **Login**. You should be able to login to IBM Robotic Process Automation Control Center successfully.



16. Once you log into control center successful, you will get below interface. The control center is the console in IBM RPA where you can manage and audit resources related to your RPA environment, such as schedules bots, create dashboards, manage computers and credentials, and configure your environments. Please refer to [RPA Control Center Interfaces](#) to get more details about its features and usage.

IBM Robotic Process Automation

Welcome to IBM Robotic Process Automation, admin!

Scaling your automation from simple back-office tasks to handling complex time-consuming business processes can be a challenge.

IBM Robotic Process Automation lets you automate your business and IT processes at scale with ease and speed. Software robots can act on AI insights to complete tasks with no lag time and accelerate your digital transformation.

Join the community

Browse the documentation

Get inspired

Get help

English (US)

4 Build it yourself – Step-by-step instructions

IBM RPA provides REST APIs for other applications to start bots. In this exercise, you will learn how a Workflow process activity can call an RPA bot to automate a swivel-chair task so far performed by a human. It will take about 1:20 hour to complete this exercise.

In the sequence of the scenario flow, it is assumed that the bot script is created first or is already available in the enterprise. You will model the business process and modify the implementation and data mapping accordingly to call the bot from the Process. In this exercise, you can use your script if you have performed the **Application Automation Using IBM RPA** lab. Or you can use the **ClientManagement** script, which has already been published to the tenant.

The IBM RPA Toolkit has been provided and deployed on the IBM Cloud Pak for Business Automation environment running on IBM Red Hat OpenShift Kubernetes Service. The toolkit provides various functionalities, including a data model that the client onboarding application uses and various service flows to start bots configured to run by an orchestrator process via server-side Asynch APIs.

This lab contains two exercises that replicate the two steps required to start bots configured to run by an orchestrator process via server-side Async APIs. The first exercise is to create an orchestration process in the IBM RPA control center. The second exercise is to call RPA server-side APIs to create an orchestration process instance that will automatically start the bot execution.

4.1 Create the RPA Orchestration Process

The server-side API uses a so-called orchestration process in the RPA server to execute the bot script. You must create and configure the orchestration process before you can call the API to request the execution of the bot. In this exercise, you will learn how to create and configure an orchestration process that contains the following three steps,

1. Create a script in the RPA studio and publish it to the RPA server
2. Create queues in the RPA control center
3. Create an orchestration process

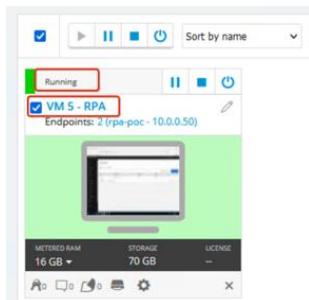
4.1.1.1 Create and Publish Your Script

Bots in IBM RPA are scripts developed using IBM RPA Studio. Please refer to [Script Development](#) to learn how to use Studio to develop the bot script. In this exercise, you can use your script if you have performed the **Application Automation Using the IBM RPA** lab. Or you can use the **ClientManagement** script published into the RPA server already.

4.1.1.2 Create a Queue

1. Click the desktop access link from the mail you received from Technology Zone and open your environment if not yet done so. When you are prompted to enter the environment password, please enter the desktop password, which is also included in

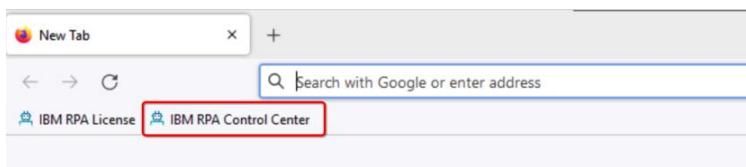
the reservation email. Once your environment is in running status as below, click **VM 5 – RPA** to open your environment.



2. Start the **Firefox** browser by clicking the Firefox icon in the Windows toolbar.



3. Click **IBM RPA Control Center** in the Firefox bookmark toolbar to launch the Control Center. You will first be taken to the login page.



4. Enter admin@rpa-poc.com as **User name**, then click **Continue**.

A screenshot of the IBM RPA Control Center login page. The page title is "Log in to IBM Robotic Process Automation". A text input field labeled "User name" contains the value "admin@rpa-poc.com", which is highlighted with a red box. Below the input field is a blue "Continue" button, also highlighted with a red box.

5. Tenant will be set to **rpa-poc** by default automatically, enter **passw0rd** (make sure to use zero as part of the password) as the password, and click **Login** to log in to the IBM RPA Control Center.

A screenshot of the IBM RPA Control Center login page. The page title is "Log in to IBM Robotic Process Automation". It asks to "Choose a tenant and enter your password". A dropdown menu under "Tenant" is set to "rpa-poc", which is highlighted with a red box. Below it, a password input field contains "passw0rd", also highlighted with a red box. At the bottom of the form is a blue "Login" button, which is highlighted with a red box.

- Click **Queues** in the left panel. Then click **Create queue**.

The screenshot shows the 'Define queues' section of the IBM RPA Control Center. The left sidebar has 'Queues' highlighted with a red box. The main area shows a table with one entry: 'InputQueue' under 'Name', 'System Queue Provider' under 'Queue provider', 'admin' under 'Modified by', and '05/18/2022' under 'Modified'. A red box highlights the 'Create queue' button at the top right of the table header.

- Configure the queue as below. Once done, click **Create**.

Queue provider: Select **System Queue Provider**.

Name: Enter a unique name; for example, prefix with your user name: **Usrxxx InputQueue**.

Description: Enter a description.

The screenshot shows the 'Create queue' dialog box. It has fields for 'Queue provider' (set to 'System Queue Provider'), 'Name' (set to 'Usrxxx InputQueue'), and 'Description' (set to 'Input Queue'). The 'Create' button at the bottom is highlighted with a red box.

4.2 Create Process to Invoke RPA Bot Asynchronously

This exercise will continue in the IBM RPA control center.

- Click **Workflows** in the left panel in the RPA control center. Select the **Processes** tab, and then click **Create Process**.

The screenshot shows the 'Manage workflows' section of the IBM RPA Control Center. The left sidebar has 'Workflows' highlighted with a red box. The main area shows a table with one entry: 'UpdateBackendSystems' under 'Process name', 'Singular/Plural' under 'Label for instances', 'admin' under 'Modified by', and '05/18/2022' under 'Modified'. A red box highlights the 'Create process' button at the top right of the table header.

- Configure the Process **General** information as below. Click **Next** once done.

Process Name: Enter a unique name to identify the Process; for example, prefix it with your user name **UsrxxxUpdateBackendSystem**.

Description: Enter a description of your Process.

Label for Singular: Enter the name to identify a single process instance; for example, **singular**

Label for Plural: Enter the name to identify multiple process instances; for example, **plural**

Create process

- General
- SLA Configuration
- Steps
- Variables

Process name
UsrxxxUpdateBackendSystem

Description
update back end systems

Select a workflow to be created with each process instances (optional)

Workflow
Workflow

Workflow Version
Select a process

What may the instances be called?

Label for Singular
singular

Label for Plural
plural

Next

- Configure the process **SLA** settings as below. Click **Next** once done.

Target Waiting Time: Enter the expected average time interval for a process instance to wait at most to start being processed, for example, 00:02:00.

Waiting Time Required Service Level: Enter the minimum acceptable percentage of instances required to meet this target, for example, 50%.

Target Handling Time: Enter the expected average time it should take a process instance to finish after it got started, for example, 00:02:00.

Handling Time Required Service Level: Enter the minimum acceptable percentage of instances required to meet this target, for example, 50%.

Target Processing Time: Enter the expected average time interval, including the waiting time for a process instance to finish the entire Process, for example, 00:04:00.

Process Time Required Service Level: Enter the minimum acceptable percentage of instances required to meet this target: 50%.

Create process

General

SLA Configuration

Steps

Variables

What is the desired time interval for an instance to wait to start being processed? What is the minimal acceptable percentage of instances that is required to meet this target?

Target Waiting Time
00:02:00

Waiting Time Required Service Level
50%

What is the desired time interval for an instance to finish the whole process after starting? What is the minimal acceptable percentage of instances that is required to meet this target?

Target Handling Time
00:02:00

Handling Time Required Service Level
50%

What is desired time interval, including waiting time, for an instance to finish the whole process? What is the minimal acceptable percentage of instances that is required to meet this target?

Target Processing Time
00:04:00

Processing Time Required Service Level
50%

[Previous](#) [Next](#)

4. Configure the process **Steps** as below. Click **Next** once done.

Step Name: Enter the step name, for example, **Update Backend Systems**.

Input Queue: Select the input queue you created in the [Create a Queue](#) section.

Output Queue (On Success): Select **Mark as success**.

Priority on Success Queue: Select **Normal**.

Output Queue (On Error): Select **Mark as error**.

Priority on Error Queue: Select **Normal**.

Script: You can select your script if you have performed the **Application Automation Using IBM RPA** lab or the **ClientManagement** script published to the server.

Version: Select the latest script version published to the server.

Computers: Select **rpa-poc** as this is the only computer configured in your RPA environment to execute bots.

Create process

General

Steps

Variables

Update Back End Systems

Step name: **Update Back End Systems**

You need to select an input queue for items to be processed and optionally a success and error queue to move items upon completion.

Input Queue: **InputQueue**

Output Queue (On Success): **Mark as success**

Priority on Success Queue: **Normal**

Output Queue (On Error): **Mark as error**

Priority on Error Queue: **Normal**

Which script will process the items in the input queue?

Script: **ClientManagement**

Version: **1 - "v1.0"**

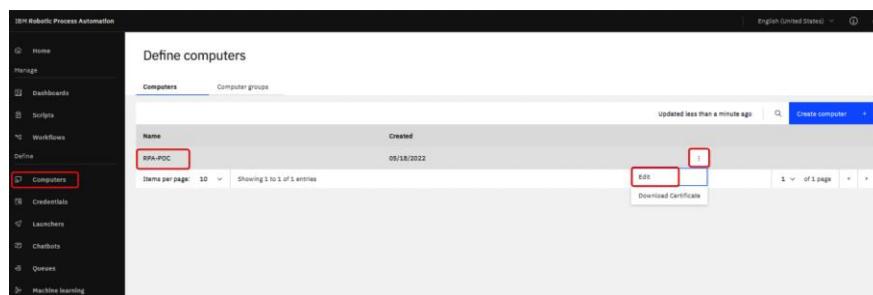
Where will this script run?

Computers: **Computers**

Next

Notes:

1. Suppose your environment has multiple computers, and you have selected them in the steps configuration, then for each message in the input queue. In that case, the bot will be started on one of the computers, providing it has an available runner license.
2. To check if your computer has allocated a certain amount of runner license for an orchestrator process to execute the bot script, click **Computers** from the left panel, click the three-dot button next to your computer, then click **Edit**.



3. The **capacity** number at the bottom is the number of total runner licenses allocated to this computer. The **Queues runtime percentage** represents the percentage of total runner licenses allocated to execute the orchestrator process. The number of runner licenses allocated for an orchestrator process equals capacity multiply queue runtime percentage, it must be equal or greater than 1.

Edit computer

Name	RPA-POC
Credential	rpa-poc
Physical address	
Computer Type	Runtime Server
Vnc Password	*****
Queues runtime percentage	50%
Standing by runtimes	
Capacity	10

Cancel Save

4. It also requires credential to unlock the computer when bot executes on it. The encrypted credentials are stored in vault. IBM Robotic Process Automation provides two type of vaults, one is System Vault (or RPA Vault) which is recommended for unattended automation, another is User Vault (or RDA vault) which is recommended for attended automation. System vault will use public key to encrypt credentials and private key to decrypt credentials which are configured in the RPA control center. Please refer to [Vault](#) for details.

Edit computer

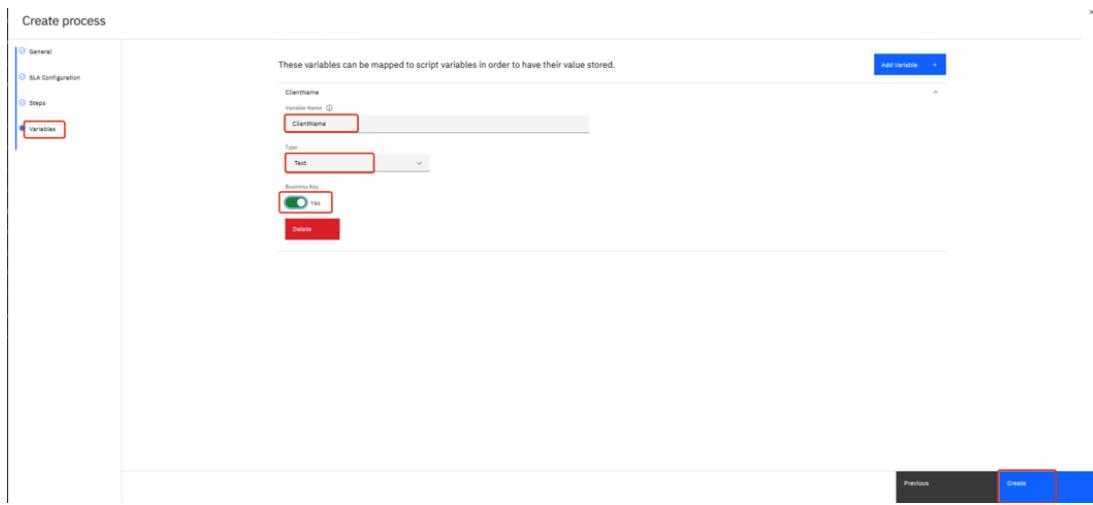
Name	RPA-POC
Credential	RPA-POC
Physical address	
Computer Type	Runtime Server
Vnc Password	*****
Queues runtime percentage	50%
Standing by runtimes	10

5. Configure **variables** as below. You can bind these variables to your scripts to persist data and track change history. During the process instance execution timeframe, you can watch the process variables' values in real-time via Control Center's dashboard. Click **Create** once done.

Variable Name: Enter the variable name, for example, ClientName.

Type: Select the variable type as Text.

Business key: set this variable as a business key. You need at least one variable set as a business key.



6. Your orchestration process should be created and listed in Manage Workflows view below.

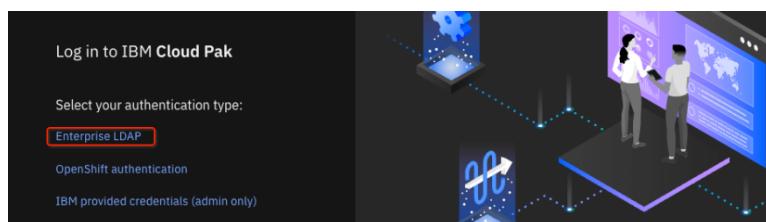
Process name	Description	Label for instances	Modified by
UpdateBackendSystems		Singular/Plural	admin
Usr001/updateBackendSystems	Usr001 Update Backend Systems	Singular/Plural	admin

4.2.1 Explore the IBM RPA Toolkit

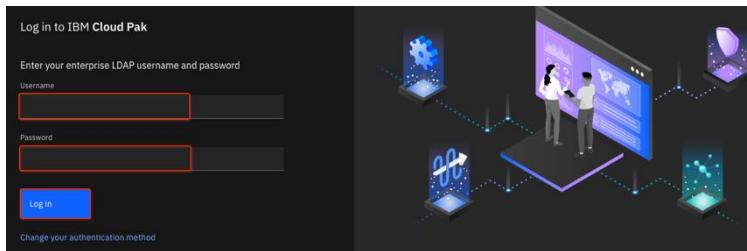
This exercise can be done from any computer as long as it has internet access to IBM Business Automation Workflow environment as part of Cloud Pak for Business Automation that is hosted on the IBM cloud.

After creating an orchestration process in RPA control center, the next step is to call the RPA server-side APIs to create the process instance, which will automatically execute the bot script in the RPA client machine. As mentioned above, to simplify this lab, a sample IBM RPA toolkit has been implemented to simplify the integration of Workflow and RPA. Before creating the workflow process, explore the IBM RPA toolkit to familiarize yourself with the IBM RPA server-side APIs.

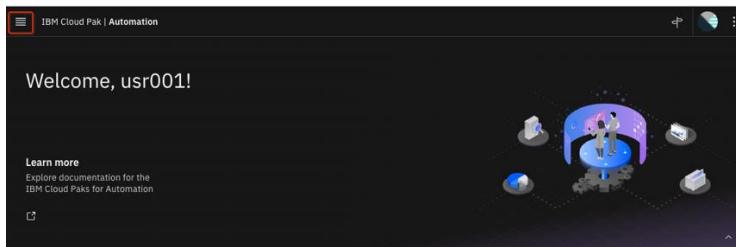
1. Log in to IBM Business Automation Studio in your web browser and select authentication type as **Enterprise LDAP**.



- Enter your IBM Cloud Pak for Business Automation username and password. Click the **Login** button to login to IBM Automation hub.



- Click the hamburger icon ☰ in the top-left corner in the IBM Automation hub and select **Design** ➔ **Business Automations** in the slideout.

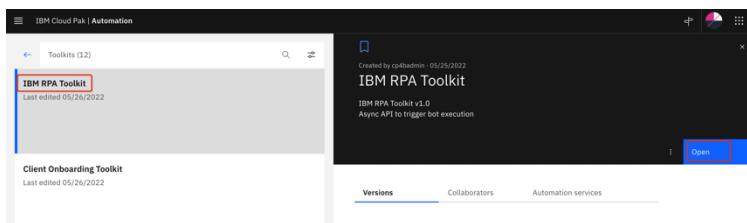


- Click **Workflow** and then click **Toolkits**. You should see the **IBM RPA Toolkit** listed on the right.

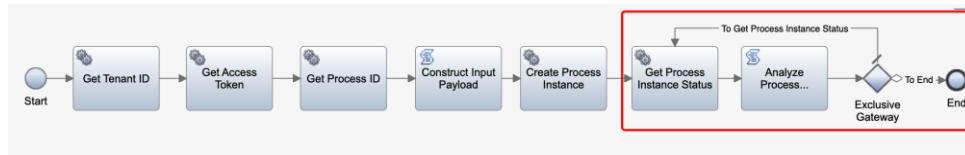
Toolkits (12)
IBM RPA Toolkit
Client Onboarding Toolkit
ADP Utils
Responsive Portal Components

- Click on the **IBM RPA Toolkit** and then click the **Open** on the right to open the toolkit in Workflow Designer.

If you see any pop-up error message window, ignore it as this is caused by a slow network and will be closed automatically.



6. The **IBM RPA Toolkit** contains two types of services that will be used in this lab. One is a data model that assigns values to the bot script's input variables. It is a string that will hold the client onboarding information business object, which will be added to the Client Management System as a JSON string. The second is a set of service flows corresponding to the RPA server-side APIs to retrieve information, including RPA tenant ID, process ID, access token etc., and create a process instance to start the RPA bot.
7. This toolkit contains 7 service flows corresponding to 7 RPA server-side APIs. You can refer to the [API Reference](#) for a detailed specification of each API. In this lab, 5 of those 7 APIs will be used, they are **Get Tenant ID**, **Get Access Token**, **Get Process ID**, **Create Process Instance**, and **Get Process Instance Status**. The typical Process to start bot via those APIs is shown below.



Once the bot starts executing after the RPA process instance is created, the Workflow process needs to query the bot execution result periodically until its execution is finished, and it also gets the output from bot.

Familiarize yourself with those service flows by clicking **Services** from the left panel and selecting the corresponding service flow,

1. **Get Tenant ID:** This is implemented using RPA API [Get user Tenants API](#). It returns the tenant ID given the tenant name and user name.

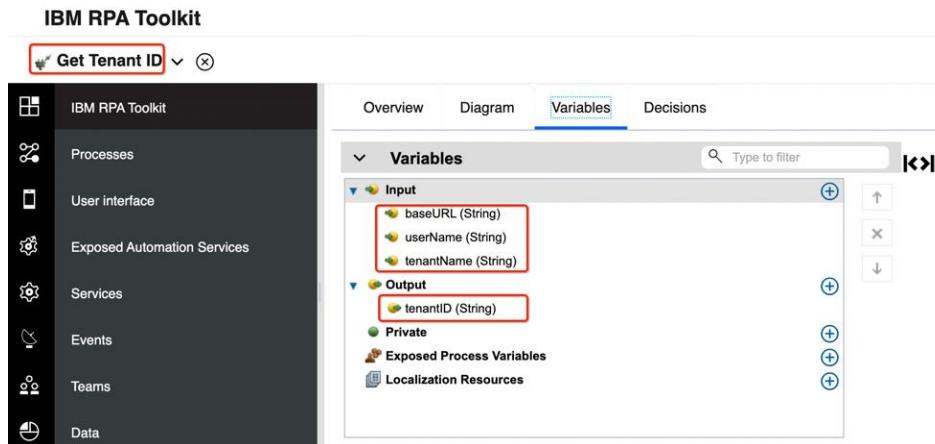
Check the input and output variables by clicking the **Variables** tab.

Input:

baseURL:	This is the base URL to access the RPA control center. Please refer to Base URLs to learn how to identify the base URL for your RPA control center.
userName:	The user's email address to get the tenant ID that the user belongs to.
tenantName:	This is the tenant name to retrieve its ID.

Output:

tenantID:	Tenant ID if the given user belongs to the given tenant or "-1" if the user doesn't belong to the tenant.
------------------	---



2. **Get Access Token**: This is implemented using RPA API [Log in to tenant](#). It returns the authorization token, which is required in all following APIs.

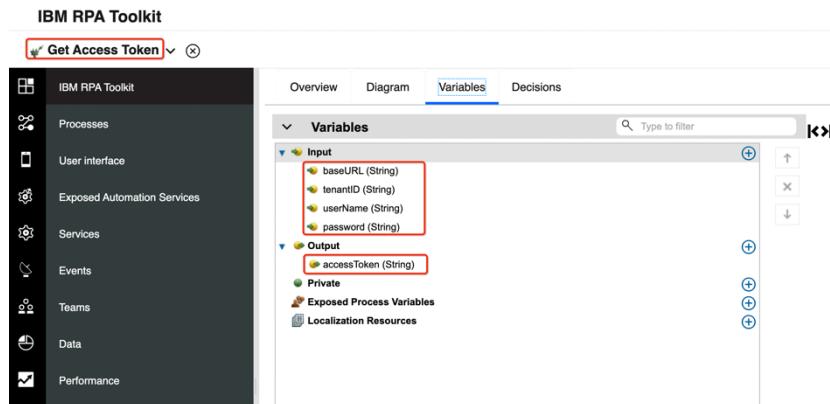
Check the input and output variables by clicking the **Variables** tab. Please note **Login in to tenant API** requires two additional input parameters, which are **grant_type** and **culture**. In this lab, we will use default grant_type which is **password**, and **en_US** for culture.

Input:

- | | |
|-------------------|---|
| baseURL : | This is the base URL to access the RPA control center. Please refer to Base URLs to learn how to identify the base URL for your RPA control center. |
| tenantID : | The ID of the tenant retrieved from Get user tenants . |
| userName : | The user's email to log in to the tenant. |
| password : | The user's password. |

Output:

- | | |
|----------------------|---|
| accessToken : | The authorization token if user logs into tenant successfully or "-1" if login fails. |
|----------------------|---|



3. **Get Process ID:** This is implemented using RPA API [Get processes](#). It returns the process ID for the specified Process defined in RPA control center.

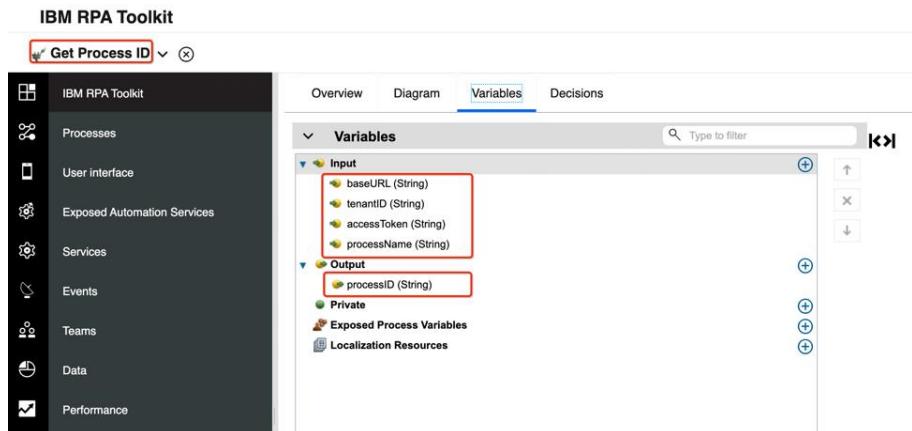
Check the input and output variables by clicking the **Variables** tab.

Input:

baseURL:	This is the base URL to access the RPA control center. Please refer to Base URLs to learn how to identify the base URL for your RPA control center.
tenantID :	The ID of the tenant retrieved from Get user tenants .
accessToken:	The authorization token retrieved from Log in to tenant .
processName:	The process name as defined in the IBM RPA control center to retrieve its process ID.

Output:

processID:	The processID of the specified Process defined in RPA control center or "-1" if the Process doesn't exist in the control center.
-------------------	--



4. **Create Process Instance:** This is implemented using RPA API [Create process instance](#). It creates a new process instance that will automatically trigger the bot execution if there is at least one available bot runner license.

Check the input and output variables by clicking the **Variables** tab.

Input:

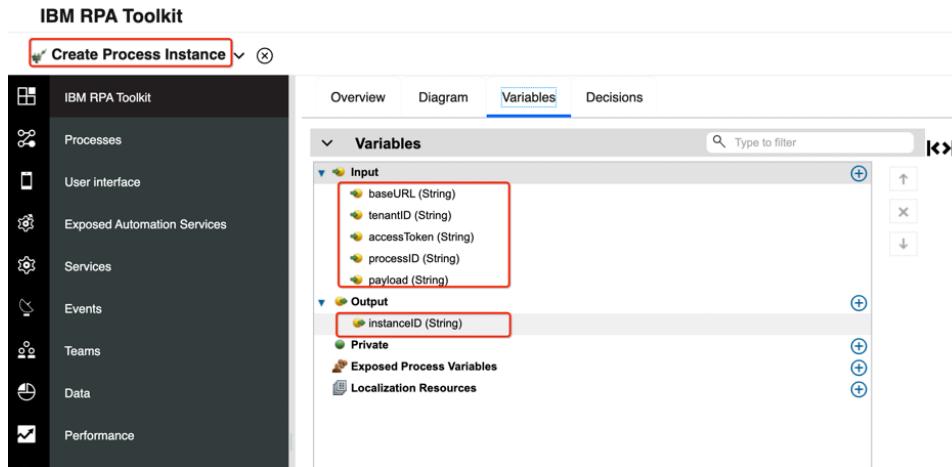
baseURL:	This is the base URL to access the RPA control center. Please refer to Base URLs to learn how to identify the base URL for your RPA control center.
tenantID :	The ID of the tenant retrieved from Get user tenants .
accessToken:	The authorization token retrieved from Log in to tenant .

processID: The process ID retrieved that retrieved from [Get processes](#).

payload: The input data passes to the script's input variables.

Output:

instanceID: The ID of the newly created process instance if a process instance was created successfully or "-1" if the creation fails.



5. Get Process Instance Status: This is implemented using RPA API [Get process instance](#). It returns the process instance result.

Check the input and output variables by clicking the **Variables** tab.

Input:

baseURL: This is the base URL to access the RPA control center. Please refer to [Base URLs](#) to learn how to identify the base URL for your RPA control center.

tenantID : The ID of the tenant retrieved from [Get user tenants](#).

accessToken: The authorization token retrieved from [Log in to tenant](#).

processID: The process ID retrieved that retrieved from [Get processes](#).

instanceID: The instance ID returned from [Create process instance](#).

Output:

instanceStatus: The process instance result as a JSON string. It contains three piece of information – **status**, **variables** and **outputs**. **status** represents the status of the process instance which can be **new**, **pending**, **processing**, **done** or **failed**.

variables represents the process instance's input variables.

outputs represents the process instance's output variables.

The screenshot shows the IBM RPA Toolkit interface with the 'Get Process Instance Status' service flow selected. The 'Variables' tab is active, showing the following variables:

- Input:** baseURL (String), tenantID (String), accessToken (String), processID (String), instanceID (String)
- Output:** instanceStatus (String)

The toolkit also contains a few other service flows, including **Get Regions**, **Get User Tenants List**, **Start RPA Bot**, and **Query Bot Execution Status**. Since they are not used in this lab, we will not explain them one by one, and you can explore them if you are interested.

4.2.2 Develop a Workflow Process to start an RPA Bot

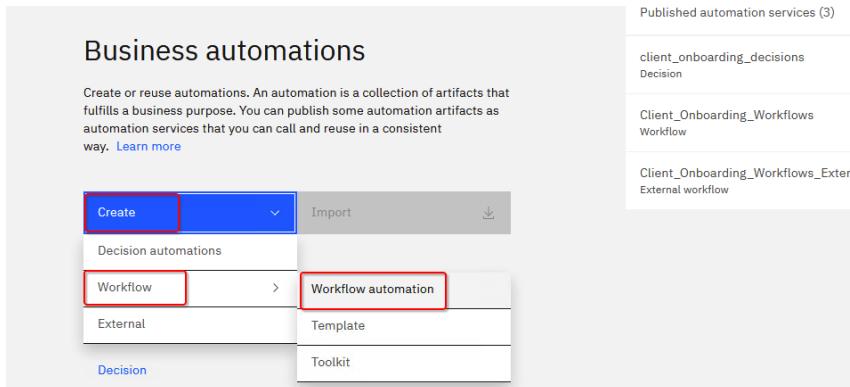
The entire end-to-end client onboarding solution involves many components: automation application, content management, automation decision service, and mobile capture. You can refer to the other labs to learn how to develop other parts of the client onboarding solution. The goal here is to showcase how activity in a Workflow process can call an RPA bot to add the client onboarding information to the backend applications. Therefore we will create a simplified process to illustrate how to call the RPA bot using the external services introduced above.

1. Click the hamburger icon in the top-left corner from Workflow Designer and select **Design**→**Business automations**.

The screenshot shows the Client Onboarding Toolkit settings page. The 'Common' section displays the toolkit's name as 'Client Onboarding Toolkit'. The 'Exposed Items' section lists the following items:

- Processes (0): The team specified in the Expose to start setting can launch instances of these processes.
- <none>
- Heritage Human Services (0): The team specified in Expose to start can view and use these services.
- <none>
- Client-Side Human Services (2)

- Click the **Create** button and select **Workflow** → **Workflow automation**.



- Leave the checkbox "Includes case features" unchecked. Enter a **name** for the workflow automation. You are using a shared environment where multiple users create business automation. To avoid conflicts and ensure that your workflow automation has a unique name, enter "**UsrXXX BAWCallRPA**" where **UsrXXX** is the username assigned to you, e.g., **Usr001 BAWCallRPA** in the workflow automation's name. For **Purpose**, although this is optional, it is recommended to enter some text to describe the purpose of this workflow automation. Once done, click **Create**.

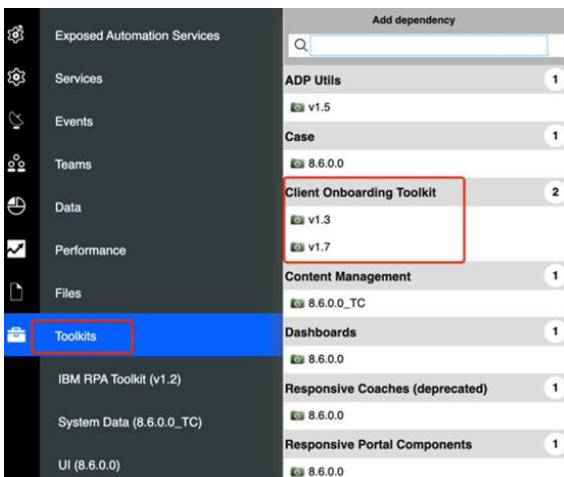
The screenshot shows the 'Create a workflow automation' dialog. It includes fields for 'Includes case features' (unchecked), 'Name' (containing 'UsrXXX BAWCallRPA'), and 'Purpose (optional)' (containing 'showcase how BAW process to start RPA bot using IBM RPA REST API'). A red box highlights the 'Purpose (optional)' text area. At the bottom right is a 'Create' button (highlighted with a red box).

- To use the data model and external services from the **IBM RPA toolkit**, the toolkit needs to be added as a dependency. Click the icon on the right next to the **Toolkits** label. Then click on the latest version of the IBM RPA Toolkit to add it as a dependency.

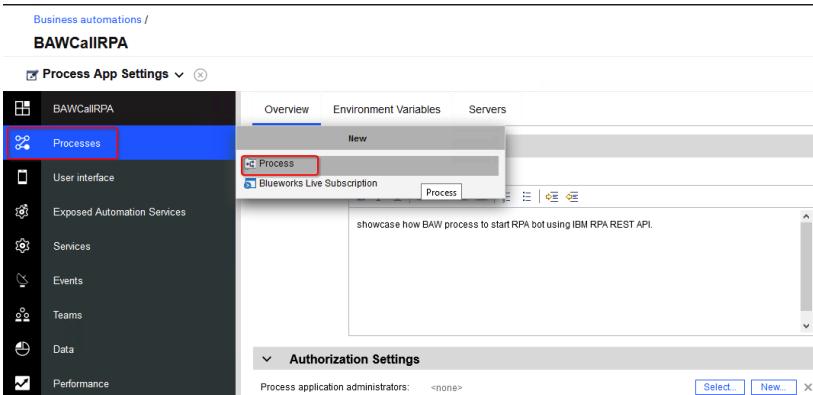
Notes: The version number may be different. Please always select the latest version/version with the highest version number.



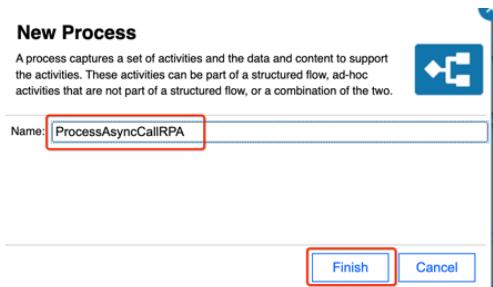
5. Also, add the **Client Onboarding Toolkit** as a dependency to use the client onboarding data model. Following the same approach, click the **+** icon on the right next to the **Toolkits** label, then click on the latest version of the Client Onboarding Toolkit to add it as a dependency.



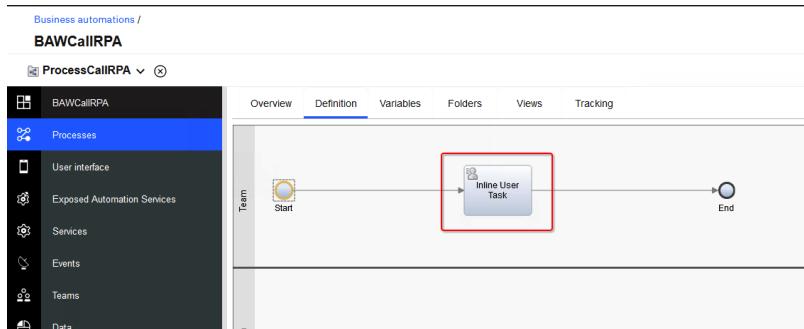
6. Click the **+** icon on the right of the **Processes** label and then click **Process**.



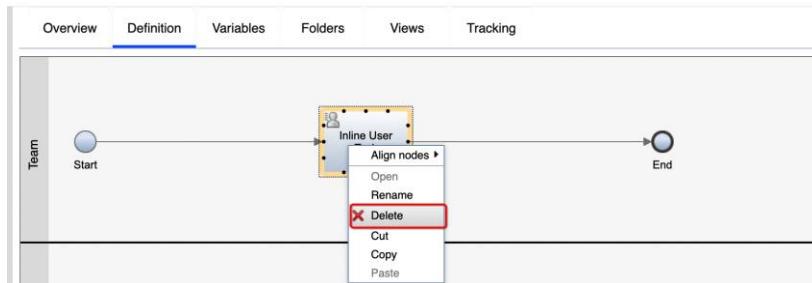
- Enter a name for the new Process, for example – ProcessAsyncCallRPA, then click **Finish**.



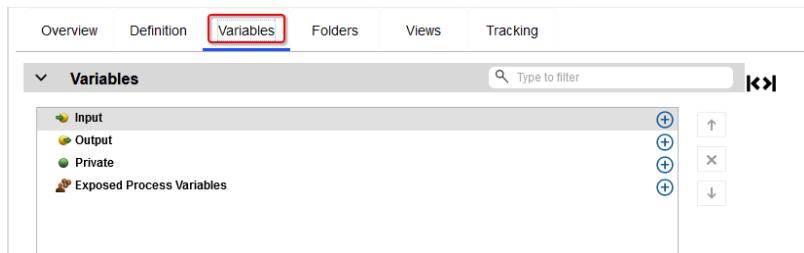
The newly created ProcessAsyncCallRPA process is opened in Workflow Designer. It initially contains one inline user task. We will change its implementation to start the RPA bot through the service flows provided in the IBM RPA Toolkit.



- Right-click the Inline User Task and select **Delete** to remove it.



- Click on the **Variables** tab to switch to the Variables view.



- Click the icon to add a **Private** variable. Change its name to **onboardingInfo** and change its type to **OnboardingInformation** defined in the Client Onboarding Toolkit. Check **Has default**, automatically generating JavaScript to construct the business object and set the default values.

The screenshot shows the 'Variables' tab in the IBM RPA Studio interface. A new private variable 'onboardingInfo' has been added, highlighted with a red box. The 'Default' section contains the following auto-generated JavaScript code:

```

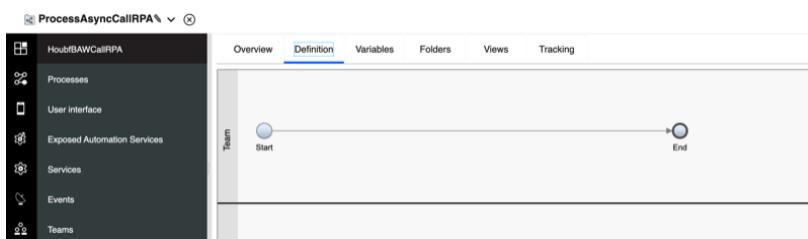
1 var autoObject = new tw.object.toolkit.COTK.OnboardingInformation();
2 autoObject.client = new tw.object.toolkit.COTK.Client();
3 autoObject.client.name = "";
4 autoObject.client.primaryContact = new tw.object.toolkit.COTK.PrimaryContact();
5 autoObject.client.primaryContact.firstName = "";
6 autoObject.client.primaryContact.lastName = "";
7 autoObject.client.primaryContact.email = "";
8 autoObject.client.primaryContact.phoneNumber = "";
9 autoObject.client.address = new tw.object.toolkit.COTK.Address();
10 autoObject.client.address.street = "";
11 autoObject.client.address.unit = "";
12 autoObject.client.address.city = "";
13 autoObject.client.address.state = "";
14 autoObject.client.address.zipCode = "";

```

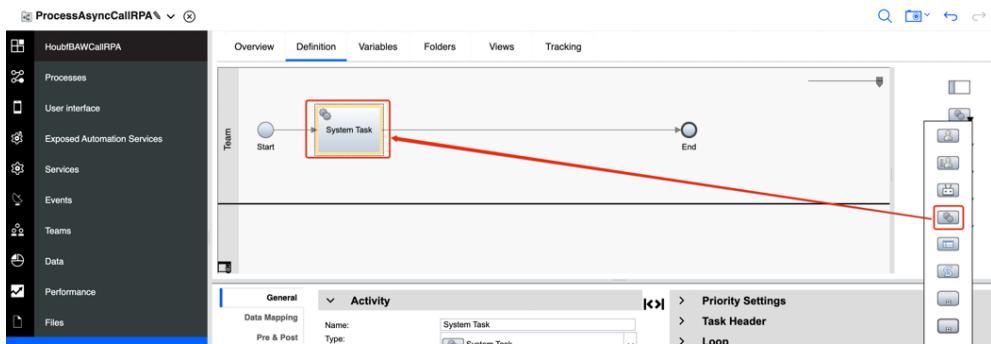
The auto-generated JavaScript constructs the business object structure and sets the default values to blank. We need to change its default value. Replace the auto-generated JavaScript code with the code from **SetDefaultValue_OnboardingInfo.js**. You can find the **SetDefaultValue_OnboardingInfo.js** on the lab materials page in the **Lab Data** folder.

- Click the icon twice next to the Private label to add two additional **private** variables. For one, change the name to **inputPayload** and select **OnboardingInfoPayload** from IBM RPA Toolkit as the type. For the other, change the name to **isProcessInstanceFinished** and select **Boolean** as the type.

- Click the **Definition** tab to switch back to the process diagram view.



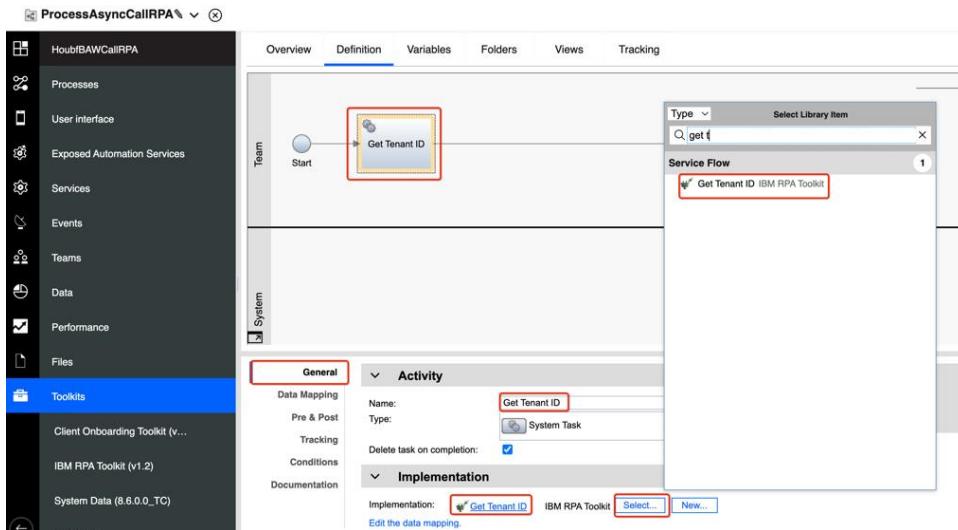
12. Drag a **System Task** activity from the right-hand palette and drop it onto the line between the **Start** and **End** activity.



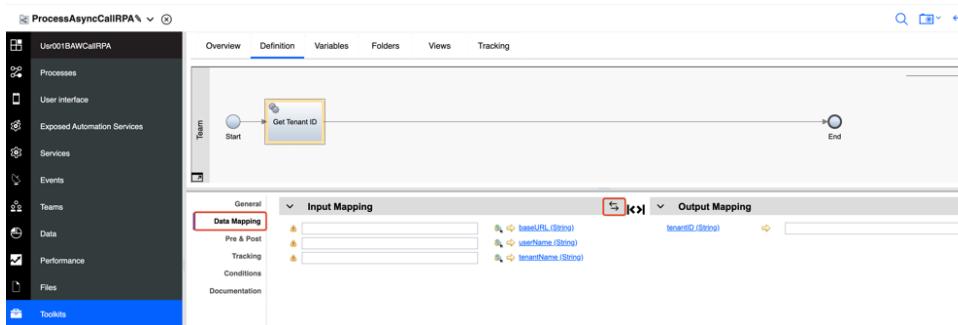
13. Configure the **System Task** activity as below:

Name: Change the name to **Get Tenant ID**.

Implementation: Click the **Select** button and select **Get Tenant ID**.



14. Switch to the **Data Mapping** tab of the activity and click the icon for Input Mapping.



15. Tick the checkbox for all three variables as **Input**. Click **Finish** to create and map the input variables automatically.

Variable Creation

Create variables where no matching variable exists. The new variables are automatically mapped. Existing mappings are not overwritten. Existing variables with the same name but different types are omitted.



Select the variables to be created and auto-mapped. By default, the variables are created as private variables. To create them as input, output, or input and output variables, select the check box beside the variable.

Variable Name	Variable Type	Input	Output
<input checked="" type="checkbox"/> baseURL	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> tenantName	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> userName	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>

< Back Next > **Finish** Cancel

16. Click the  icon for Output Mapping, and tick the checkbox for tenantID as Output. Click **Finish** to create and map the output variable automatically.

Variable Creation

Create variables where no matching variable exists. The new variables are automatically mapped. Existing mappings are not overwritten. Existing variables with the same name but different types are omitted.

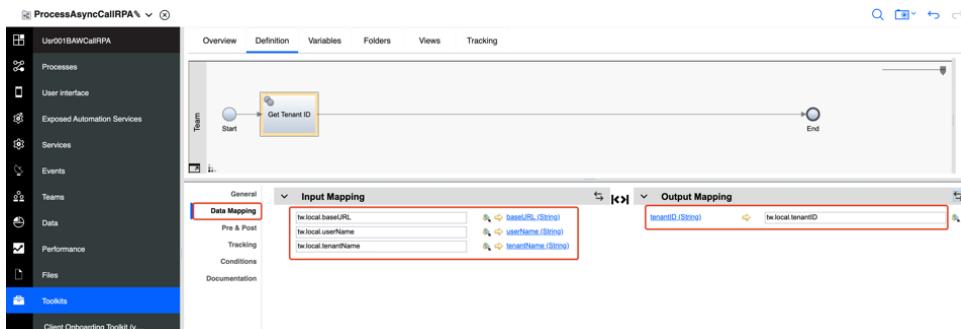


Select the variables to be created and auto-mapped. By default, the variables are created as private variables. To create them as input, output, or input and output variables, select the check box beside the variable.

Variable Name	Variable Type	Input	Output
<input checked="" type="checkbox"/> tenantID	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>

< Back Next > **Finish** Cancel

17. Once done, the **Data Mapping** for the **Get Tenant ID** system task should look as shown below.

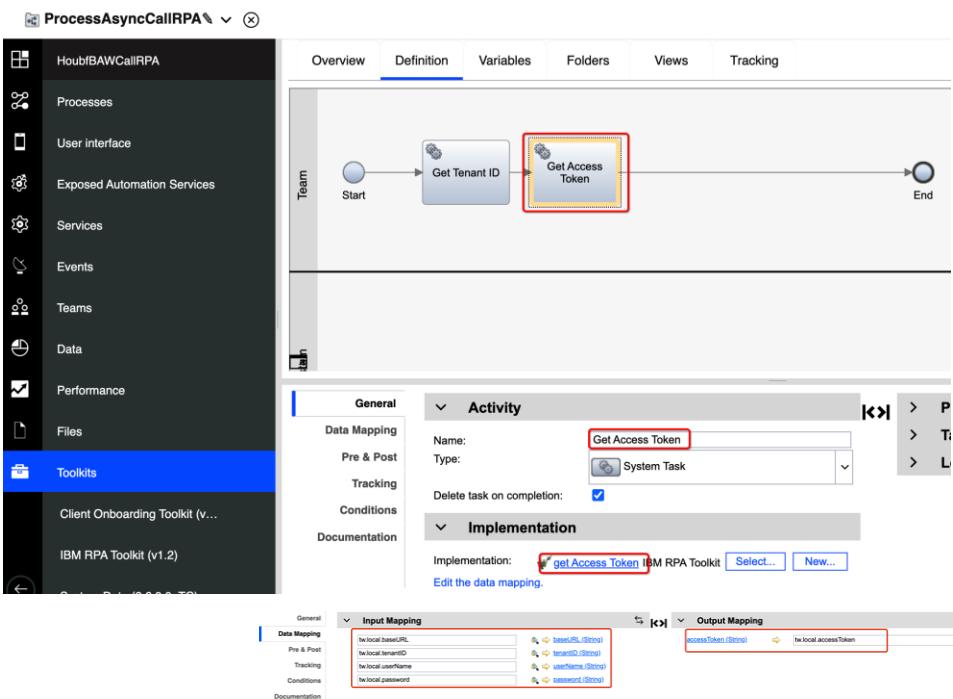


18. Follow the same approach to add a second system task and configure it as below:

Name: Change its name to **Get Access Token**

Implementation: Select the **Get Access Token** service flow as its implementation.

Data Mapping: Click for both Input and Output mapping to automatically create and map the Input and Output variables.

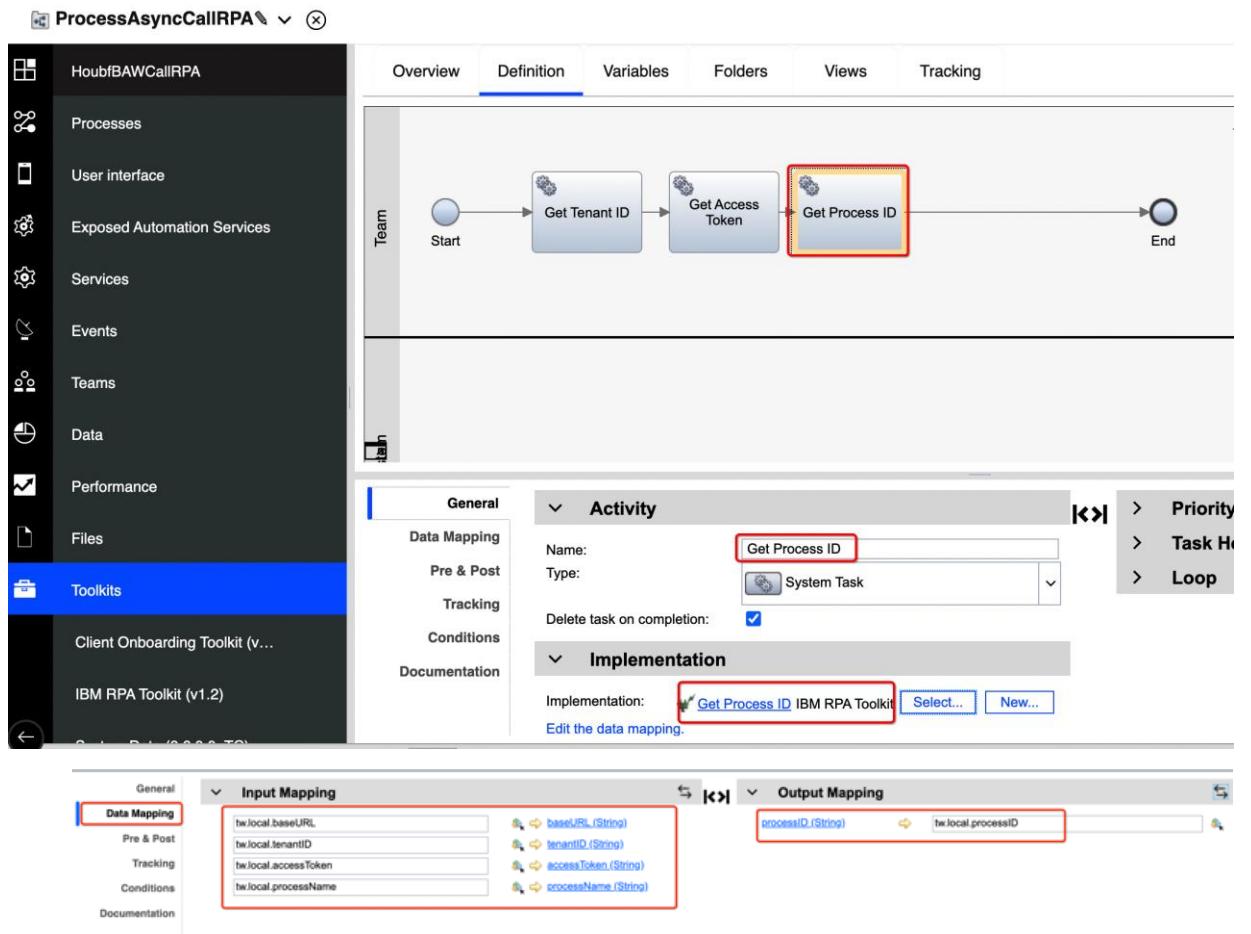


19. Follow the same approach to add a third system task and configure it as below:

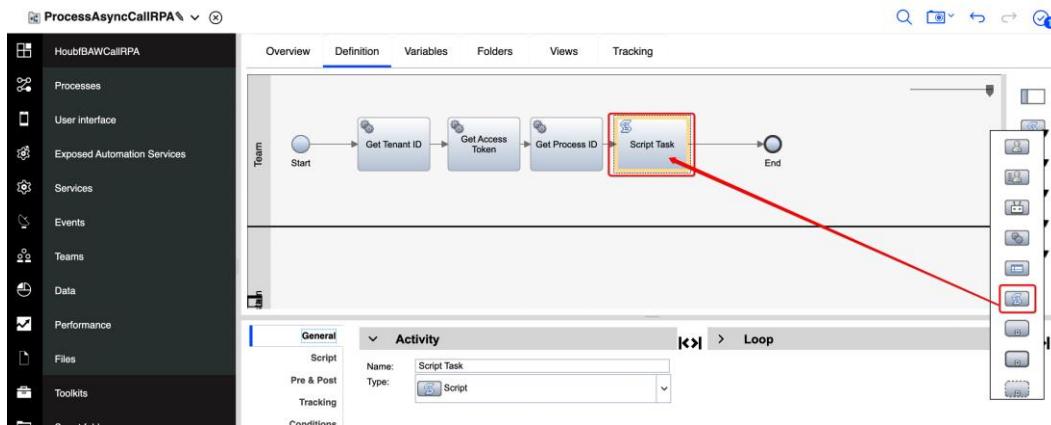
Name: Change its name to **Get Process ID**

Implementation: Select the **Get Process ID** service flow as its implementation.

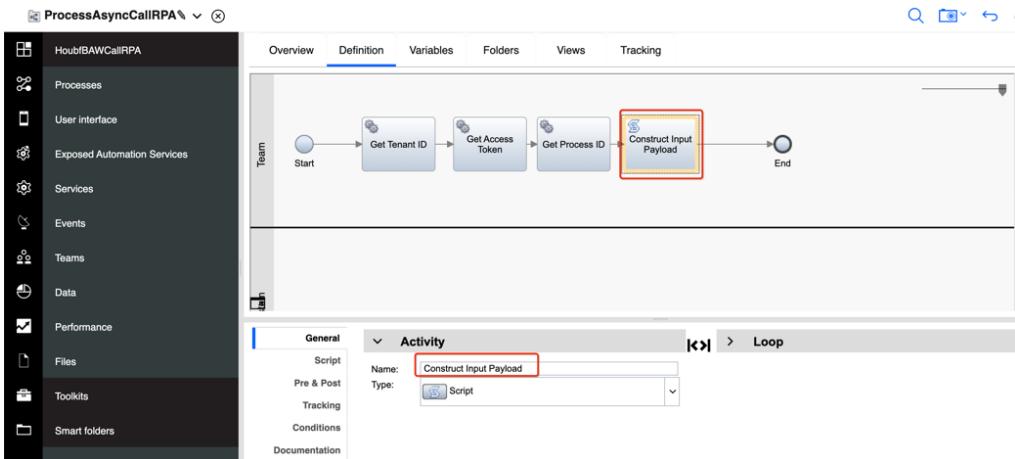
Data Mapping: Click for both Input and Output mapping to automatically create and map the Input and Output variables.



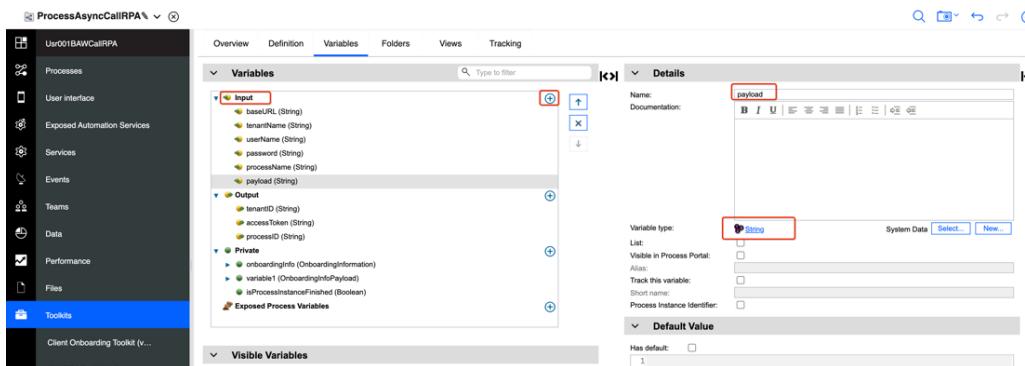
- Before we can call the **Create Process Instance** API to create a new process instance on the RPA control server to execute the bot, we need to construct the bot input payload. The client onboarding information is stored in the variable **onboardingInfo**, a business object. We need to convert it to a JSON string and pass that to the bot as a payload. Drag a **Server Script** activity and drop it onto the line between Get Process ID and End activity as below.



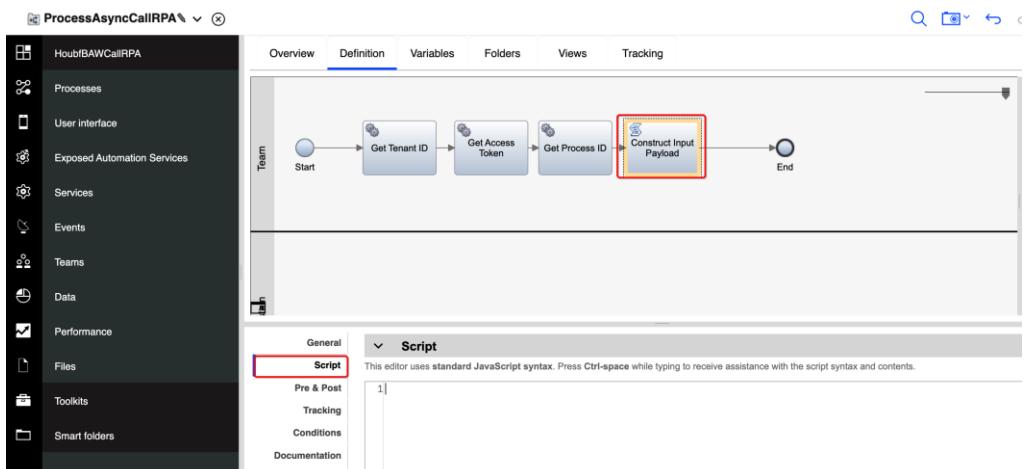
21. Change the name of the Server Side script activity to **Construct Input Payload**.



22. Click the **Variables** tab to switch to the Variables view. Click the **+** icon next to **Input** to add a new input variable. Name it **payload** and select **String** as its type.

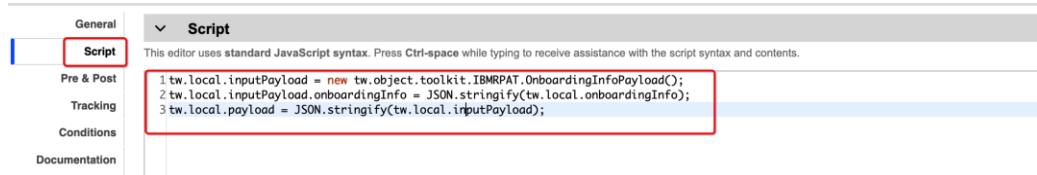


23. Click the **Definition** tab to switch back to the process diagram definition. Select the **Construct Input Payload** server-side script activity.



24. Click the **Script** tab of the activity to switch to the Script view, and copy and paste below the JavaScript snippet into the Script editor. This creates an instance of the OnboardingInfoPayload business object from the RPA integration toolkit, sets the JSON representation of the onboardingInfo variable in the instance, and assigns the JSON representation to the payload variable.

```
tw.local.inputPayload = new tw.object.toolkit.IBMRPAT.OnboardingInfoPayload();
tw.local.inputPayload.onboardingInfo = JSON.stringify(tw.local.onboardingInfo);
tw.local.payload = JSON.stringify(tw.local.inputPayload);
```

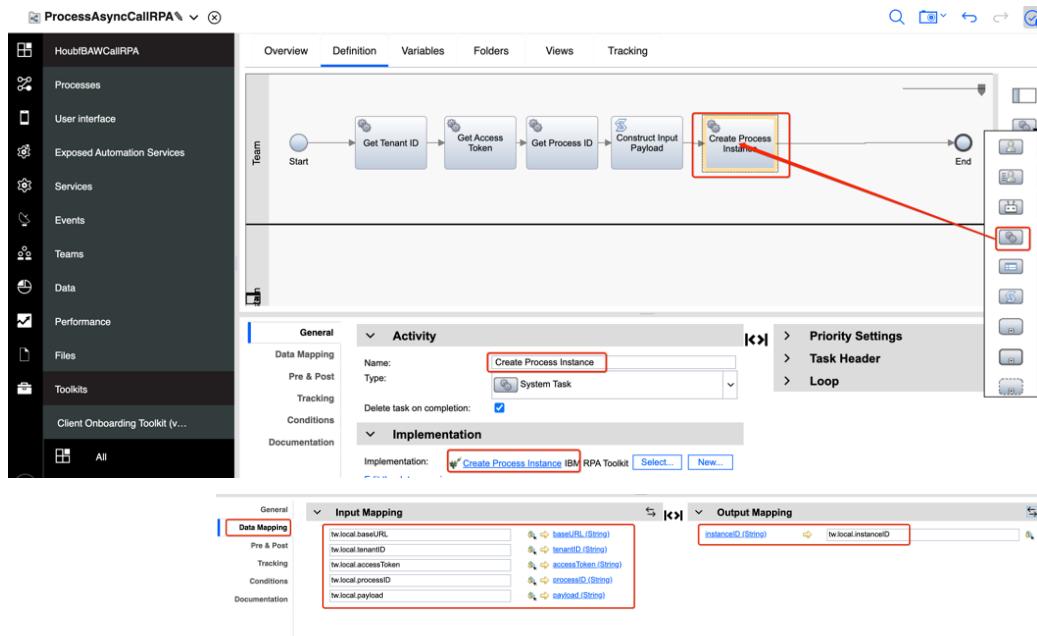


25. Drag a **System Task** activity and drop it onto the line between the **Construct Input Payload** and **End** activities, and configure it as below:

Name: Change its name to **Create Process Instance**.

Implementation: Select the **Create Process Instance** service flow

Data Mapping: Click ↗ for Input and Output mapping to automatically map Input and Output variables.

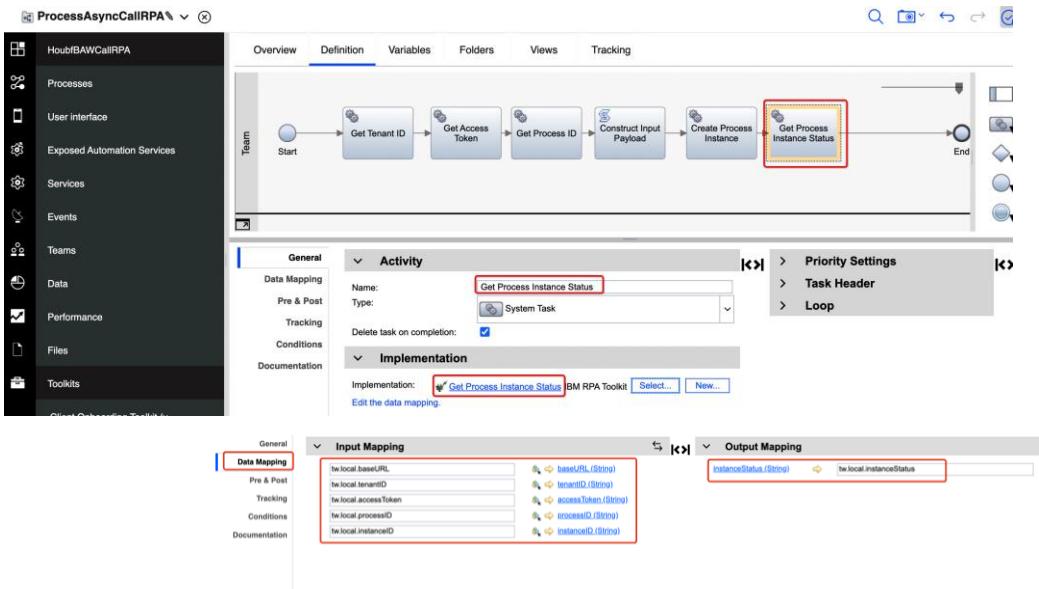


26. After the process instance has been created in the RPA control server, we need to check if it has been processed or not periodically. This can be achieved using the **Get Process Instance Status** API. Add another **System Task** activity to query process status, and configure it as below:

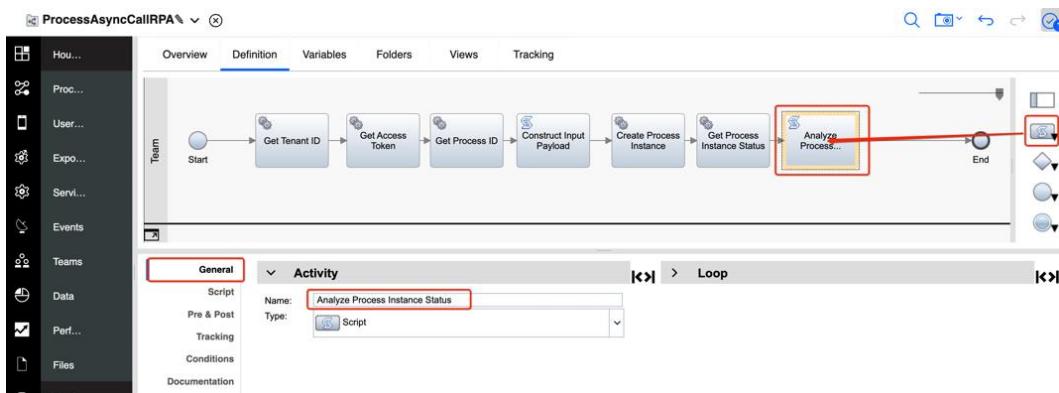
Name: Change its name to **Get Process Instance Status**.

Implementation: Select the **Get Process Instance Status** service flow.

Data Mapping: Click ↳ for both Input and Output mapping to automatically map Input and Output variables.

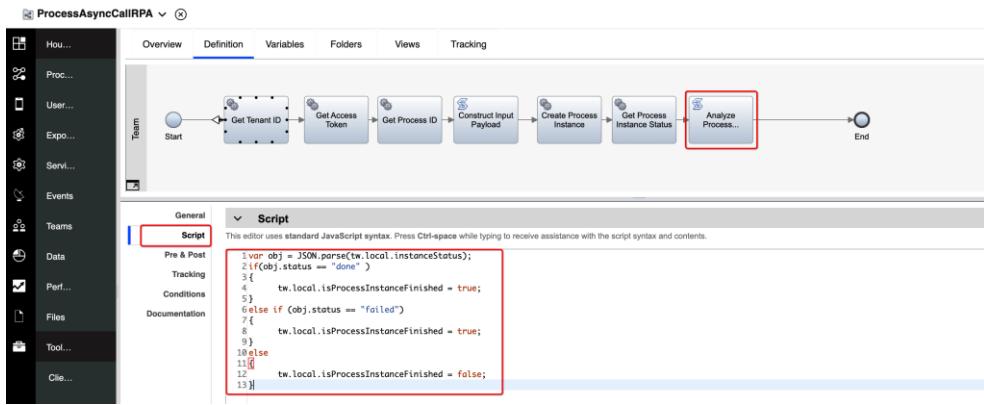


- Once we get the process instance status, we need to check if the process instance has finished processing or not. Drag a server-side script activity and drop it onto the line between the **Get Process Status** and **End** activities. Change its name to **Analyze Process Instance Status**.

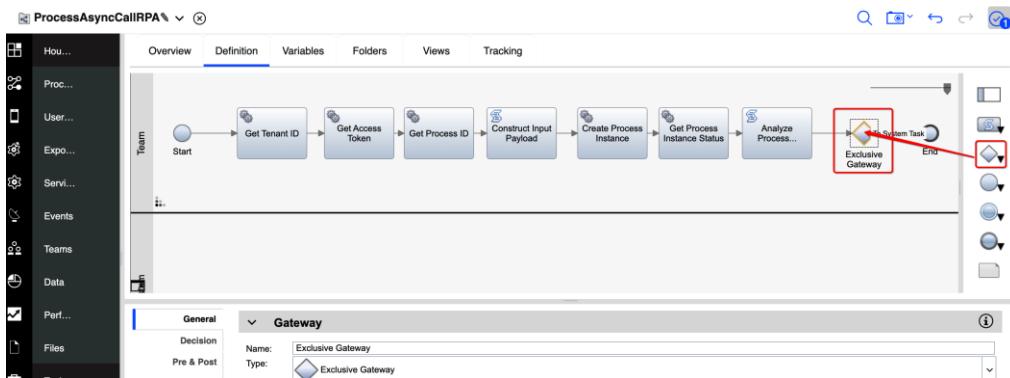


- Click **Script** to switch to Script tab, copy and paste below JavaScript snippet into script editor.

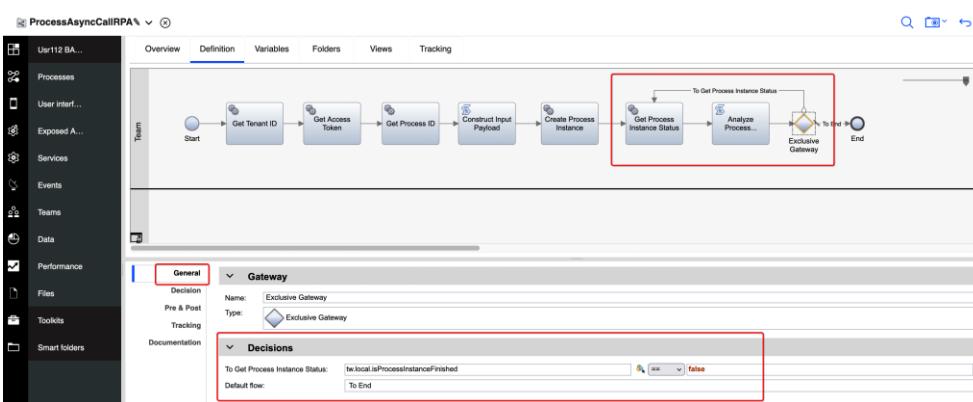
```
var obj = JSON.parse(tw.local.instanceStatus);
if(obj.status == "done")
{
    tw.local.isProcessInstanceFinished = true;
}
else if (obj.status == "failed")
{
    tw.local.isProcessInstanceFinished = true;
}
else
{
    tw.local.isProcessInstanceFinished = false;
}
```



29. If the process instance on the RPA control server has finished processing, the Workflow process will continue to the **End** activity. If it is still being executed the Workflow process needs to go back to the **Get Process Instance Status** activity to recheck the process instance status. Drag and drop a **Gateway** activity onto the line between the **Analyze Process Instance Status** and **End** activities.



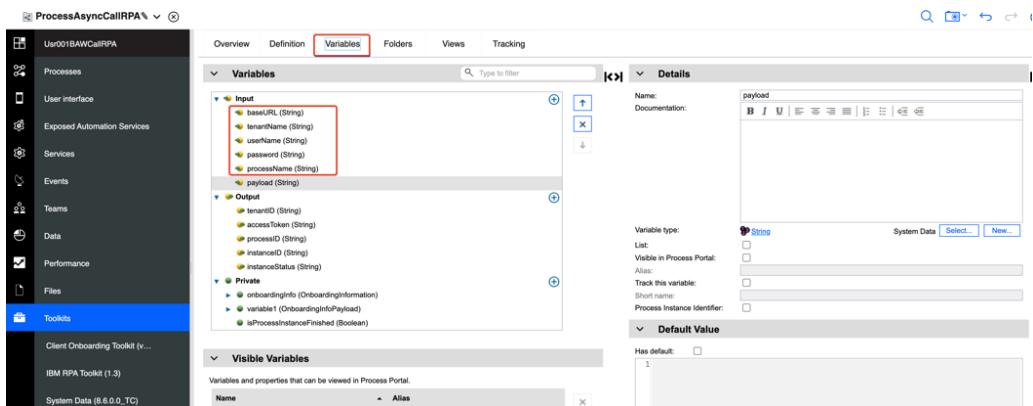
30. Connect the **Gateway** with the **Get Process Instance Status** activity and set the gateway **Decisions** to **tw.local.isProcessInstanceFinished == false**, as shown below:



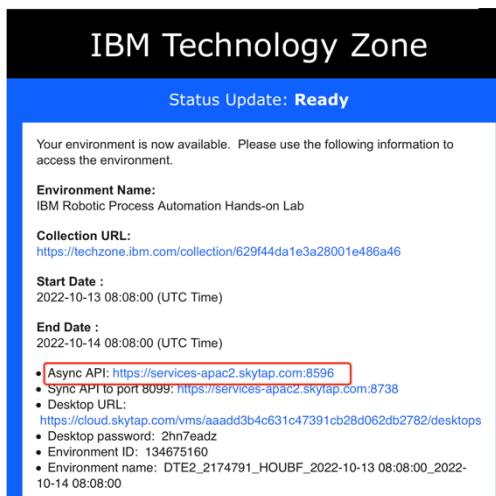
31. Your Process should be similar to the one below. It contains 5 system task activities and 2 server-side script activities. Before we can test the Process, we also need to set the RPA tenant and process information.

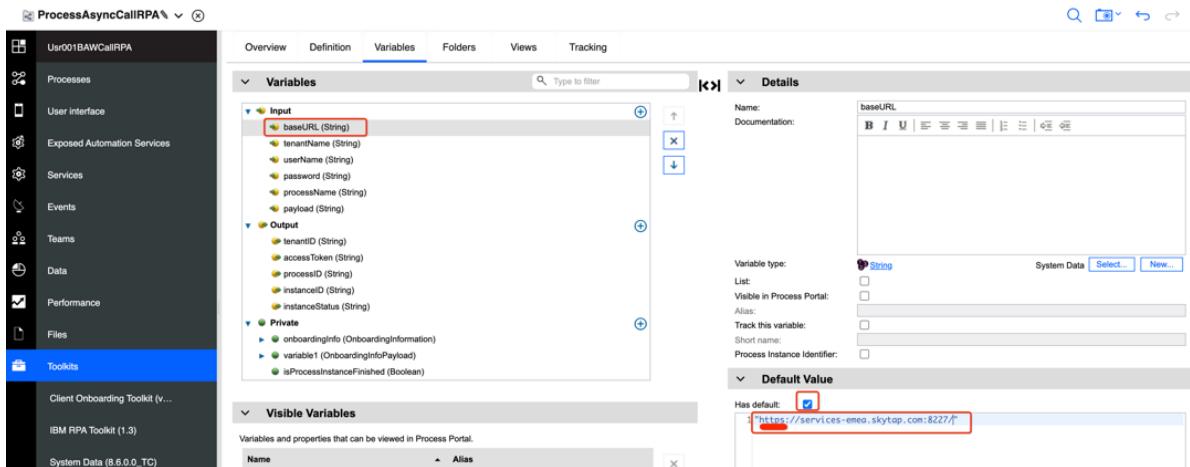


32. Click on the **Variables** tab to switch to the variables view. A couple of input variables have been created as part of mapping the input values of some of the system activities. To execute the Process, values for **baseUrl**, **tenantName**, **userName**, **password**, and **processName** need to be set.



33. Select the **baseUrl** input variable, check **Has default** in the bottom-right corner. Since the Technology Zone environment is isolated, the services running in this environment are not accessible externally. We have exposed IBM RPA REST APIs via Published Services. To get the **baseUrl** value, please go back to your mail box and find the email you received from IBM Technology Zone. **Copy** the address of the application Async API service from your reservation mail as below and paste it as the default value for the **baseUrl**.





34. Repeat the same steps to set the default value to **tenantName**, **userName**, **password**, and **processName** as below:

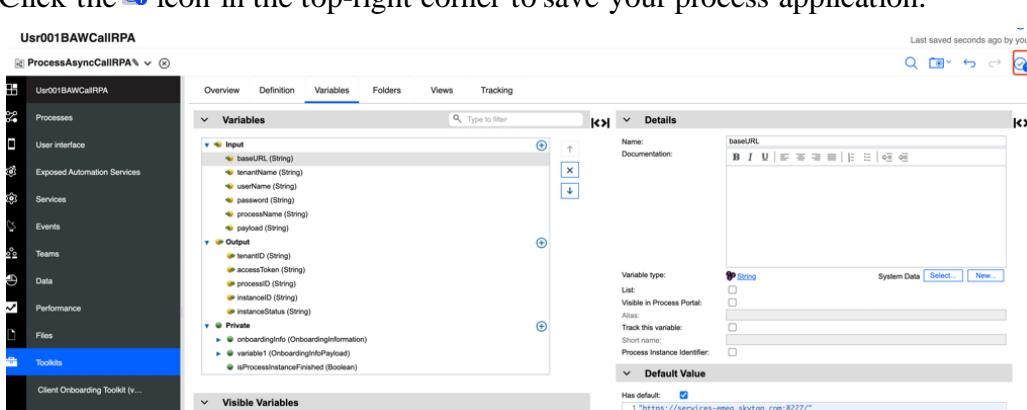
tenantName: rpa-poc

userName: admin@rpa-poc.com

password: passw0rd (make sure to use a zero as part of the password)

processName: Enter the process name you defined in the exercise [Create an orchestration process](#) or use the pre-configured process name: **UpdateBackendSystems**.

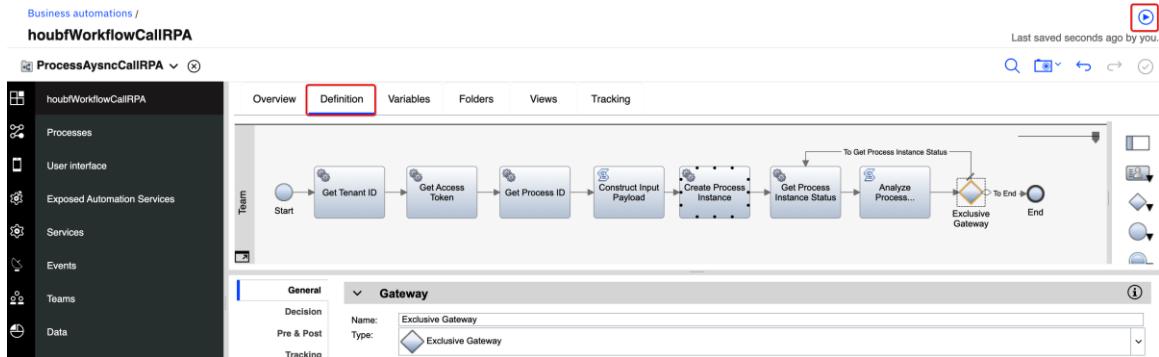
35. Click the icon in the top-right corner to save your process application.



4.2.3 Verification Instructions

Using the Playback and Inspector, you can quickly test the Process directly from the development environment without publishing it to a Workflow server. We will use it to validate if the Workflow process authored above can successfully start the bot.

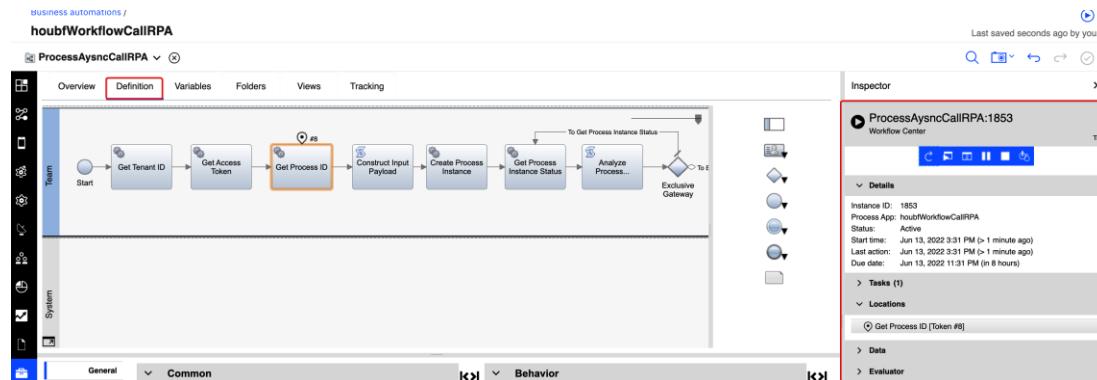
1. Click **Definition** to switch to the process diagram view, then click the icon in the top-right corner in the Workflow Designer window. It will start a new process and show it in the Inspector.



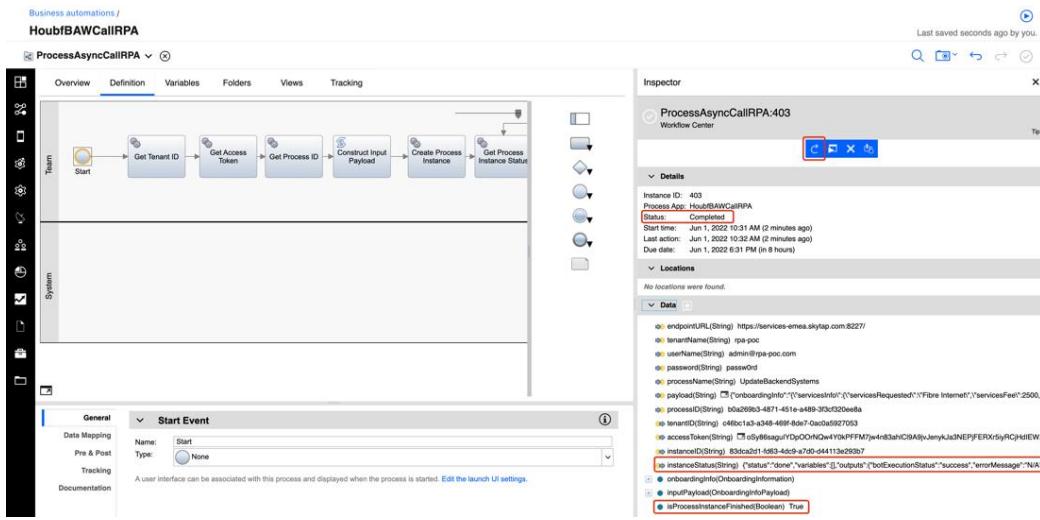
If you see a warning message indicating Firefox prevented this site from opening a pop-up window, click the **Options** and then select "Allow pop-ups for..." to allow Firefox to open a pop-up window.



2. A new process instance will be started as below. It will call the service flows that calls the RPA server-side APIs to create process instance and execute the bot script on VM 5.



3. Watch the Windows desktop on VM 5 – RPA, but don't touch the keyboard or mouse. The bot will be started. It will first start the Client Management System Java application to add the client information and grab the client ID. Next, the Service Management System web application will start to add the signed services for the client. Please return to the **Inspector** once the bot execution finishes.
4. Click the icon in the Inspector window to refresh the process instance status. You may need to refresh it several times until its status becomes **Completed**. Check the bot execution status by reviewing the value of **isProcessInstanceFinished** in the data section. It should be "true" indicating that the bot has been executed successfully. Please also check the **instanceStatus** which contains three piece of information – **status**, **variables** and **outputs**. **outputs** hold the output response from the bot.



Summary

In this exercise, you have learned how to:

- create a queue;
- create and configure an orchestration process in the RPA control center;
- implement a workflow process that calls a series of service flows that use external services to interact with the RPA control server to run a bot;
- start an RPA bot through the IBM RPA server-side Async REST API and get its status.

Congratulations, you have completed this lab!!!