

Integration with UiPath Robotic Process Automation



Note: This tutorial is for the older UiPath Studio 19.5. Please refer to the latest [Integration with UiPath Robotic Process Automation](#) for the latest UiPath Studio 20.10, which provides a much simplified [sign in](#) and [first run experience](#) compared to previous releases.

- Introduction to UiPath
 - UiPath Overview
 - UiPath Concepts and Terms
 - Get Started with UiPath
 - Step 1: Sign Up for UiPath Cloud Platform
 - Step 2: Install UiPath Studio and UiPath Robot
 - Step 3: Launch UiPath Robot and Obtain Machine Name
 - Step 4: Obtain Machine Key
 - Step 5: Connect UiPath Robot to the Machine
 - Step 6: Create a Robot
 - Step 7: Add Robot to Environment
 - Step 8: Start a Job
 - Step 9: Prepare for UiPath Integration
 - 1. Get Access and ID Tokens
 - 1.1 Generate Code Challenge and Code Verifier
 - 1.2 Get Authorization Code
 - 1.3 Get Refresh Token
 - 1.4 Get Access Token and ID using the Refresh Token
 - 2. Get UiPath Account, Service and Process Identifiers
 - 2.1 Get the Account Logical Name
 - 2.2 Get Service Instance Logical Name
 - 2.3 Get Release Key for the Desired Process
 - 2.4 Test Orchestrator API Call to Start a Job
- Integrate UiPath with Joget
 - How to Integrate with UiPath
 - Design a Process to Start a UiPath Job
 - Step 1. Design New App
 - Step 2. Design New Process
 - Step 3. Configure First Tool to Get UiPath Access Token
 - Step 4. Configure Second Tool to Start Job Using the Access Token
 - Step 5. Run the Process
 - Sample Joget App

Introduction to UiPath

UiPath Overview

UiPath is one of the leading Robotic Process Automation (RPA) vendors. UiPath consists of 3 main components:

[blocked URL](#)

UiPath Component Architecture

1. [UiPath Studio](#): UI tool to visually design the process to automate
2. [UiPath Orchestrator](#): Web application that manages the creation, monitoring, and deployment of all robots and processes
3. [UiPath Robot](#): Runs processes that were built in UiPath Studio. Execution agent that is installed and executed in the actual machine.

UiPath Concepts and Terms

Here are some of the basic concepts and terms in UiPath that you should get familiar with for the tutorial later:

- [Machine](#): Represents the actual machine the robot executes in
- [Environment](#): An environment is a grouping of Robots, that is used to deploy processes

- **Package:** A UiPath Studio project that is published
- **Process:** A process represents the association between a package and an environment. Each time a package is deployed to an environment, it is automatically distributed to all machines that belong to that environment.
- **Job:** A job is the execution of a process on one or multiple Robots.

There are other concepts for more advanced usage that will not be used in the tutorial, such as:

- **Library:** A process library describes a system of activities that are reusable for sharing
- **Schedule:** Enables jobs to be executed in a preplanned manner
- **Asset:** Usually represent shared variables or credentials that can be used in different projects.
- **Queue:** A place to store multiple types of data, such as invoice information or customer details.

Get Started with UiPath

Step 1: Sign Up for UiPath Cloud Platform

Let's sign up for an account at <https://www.uipath.com/platform-trial>. There is a Community plan to get started for free, so we'll Choose Community to register.

[blocked URL](#)

Sign up using a social account or an email, and upon registration you will be taken to a dashboard at <https://platform.uipath.com>.

[blocked URL](#)

A service represents a deployment in the company. A default service is already created e.g. DemoDefault.

[blocked URL](#)

Select the service name to open the UiPath Orchestrator web application.

[blocked URL](#)

Step 2: Install UiPath Studio and UiPath Robot

Download the UiPath Studio installer (UiPathStudioSetup.exe) from the [Resource Center](#), and install it on the target computer. More information on UiPath Studio is available at the [UiPath Studio Guide](#).

[blocked URL](#)

After installing UiPathStudioSetup.exe, start UiPath Studio from the Windows Start menu and activate it. You can start free by activating the Community Edition.

[blocked URL](#)

Step 3: Launch UiPath Robot and Obtain Machine Name

In the Windows Start menu, search for UiPath Robot and start it

[blocked URL](#)

Click on the Gear icon to open the Orchestrator Settings window, and copy the Machine Name.

[blocked URL](#)

Step 4: Obtain Machine Key

Back in the UiPath Orchestrator, create a new Machine under the Machines menu.



IMPORTANT: Make sure that the name matches the Machine Name shown in the UiPath Orchestrator Settings earlier.

[blocked URL](#)

View the selected Machine and copy the Machine Key.

[blocked URL](#)

[blocked URL](#)

Step 5: Connect UiPath Robot to the Machine

In the UiPath Robot, click on the Gear icon to open the Orchestrator Settings window.

Fill in the Orchestrator URL (<https://platform.uipath.com> for the UiPath Cloud Platform) and the Machine Key copied in the previous step, then click on Connect.



Once connected, you should see “Status: Robot unavailable” at the bottom of the window.

Hover over that message, and you should see the reason “Robot is not available.” This message means that the machine connection is valid, so now it’s time to create a robot. If you see a different reason then there might be an error in your configuration so you will want to double check your settings.

[blocked URL](#)

Step 6: Create a Robot

In the UiPath Orchestrator under the Robots menu, click on the Plus icon on the top right.

Choose “standard robot that works on a single standard machine”.

Assign the robot to the machine created previously.

Set a “Name”, and fill in the actual Windows login for the target machine (where the UiPath Robot is installed) under Domain\Username and Password.

[blocked URL](#)

Step 7: Add Robot to Environment

Next, we’ll need to add the created Robot to the Environment. Select the Environments tab at the top of the Robots menu and select the Manage option.

[blocked URL](#)

Select the previously created robot and click UPDATE.

[blocked URL](#)

At this point, the robot should be successfully connected to the machine. Check the UiPath Robot Orchestrator Settings and the status should show “Connected, licensed”

[blocked URL](#)

Step 8: Start a Job

Now that the robot is deployed, it’s time to start a job to test it.

Under Jobs, click on the Play icon on the top right.

Select the pre-existing Demo Process and the previously robot, then click on Start.

[blocked URL](#)

[blocked URL](#)

In the UiPath Robot, it will show “Installing package...”. Once the package has been installed, it will show “Job started processing” which means the process has started. In this demo process, a Hello popup dialog will appear.

[blocked URL](#)

The robot has successfully started the job. You can monitor the robots and jobs in the Monitoring menu in the UiPath Orchestrator.

[blocked URL](#)

Step 9: Prepare for UiPath Integration

In June 2019, UiPath introduced a new mechanism for [consuming its Cloud APIs](#). There are a number of steps required to retrieve some required information for integration:

1. Get Access and ID Tokens

1.1 Generate Code Challenge and Code Verifier

In a web browser, visit <https://repl.it/languages/nodejs> and run the following code:

```
function base64URLEncode(str) {
  return str.toString('base64')
    .replace(/\+/g, '-')
    .replace(/\//g, '_')
    .replace(//=g, '');
}

function sha256(buffer) {
  return crypto.createHash('sha256').update(buffer).digest();
}

console.log("Generating challenge and Verifier");
var cryptoResult = crypto.randomBytes(32);
var verifier = base64URLEncode(cryptoResult);
var challenge = base64URLEncode(sha256(verifier));
console.log("Code challenge: " + challenge);
console.log("Code verifier: " + verifier);
```

Take note of the **code challenge** and **code verifier** values in the output e.g.

```
Generating challenge and Verifier
Code challenge: RzYlHliGzPGgOLaRQJYftZlmmc3sCbeicZVRftTmC-A
Code verifier: YVMnLczXQgJ9dwzV7MlMWEjGyAia4nXvTZzU4UVrAPE
```

1.2 Get Authorization Code

Replace the **code challenge** into the URL below, and visit the updated URL in a browser.

[https://account.uipath.com/authorize?response_type=code&nonce=b0f368cbc59c6b99ccc8e9b66a30b4a6&state=47441df4d0f0a89da08d43b6dfdc4be2&code_challenge=\[code_challenge\]&code_challenge_method=S256&scope=openid+profile+offline_access+email&audience=https%3A%2F%2Forchestrator.cloud.uipath.com&client_id=5v7PmPJL6FOGu6RB8l1Y4adLBhlwovQN&redirect_uri=https%3A%2F%2Faccount.uipath.com%2Fmobile](https://account.uipath.com/authorize?response_type=code&nonce=b0f368cbc59c6b99ccc8e9b66a30b4a6&state=47441df4d0f0a89da08d43b6dfdc4be2&code_challenge=[code_challenge]&code_challenge_method=S256&scope=openid+profile+offline_access+email&audience=https%3A%2F%2Forchestrator.cloud.uipath.com&client_id=5v7PmPJL6FOGu6RB8l1Y4adLBhlwovQN&redirect_uri=https%3A%2F%2Faccount.uipath.com%2Fmobile)

The browser should redirect to:

[https://account.uipath.com/mobile?code=\[authorization_code\]&state=47441df4d0f0a89da08d43b6dfdc4be2](https://account.uipath.com/mobile?code=[authorization_code]&state=47441df4d0f0a89da08d43b6dfdc4be2)

Copy the **authorization_code** in the URL.

1.3 Get Refresh Token

Using any API testing tool (e.g. [Postman](#)), submit a POST request to

<https://account.uipath.com/oauth/token> with the body below, replacing the values for **[authorization_code]** and **[code_verifier]** accordingly.

```
{
  "grant_type": "authorization_code",
  "code": "[authorization_code]",
  "redirect_uri": "https://account.uipath.com/mobile",
  "code_verifier": "[code verifier]",
  "client_id": "5v7PmPJL6FOGu6RB8I1Y4adLBhIwovQN"
}
```

Copy the value of the **refresh_token** from the response:

```
{
  "access_token": "eyJ0eX...",
  "refresh_token": "CBZcOD6vrP2FQ9qa8fuqDdfoEwnVPuR2Kpz-hitbTAIzG",
  "id_token": "eyJ0eX...",
  "scope": "openid profile email offline_access",
  "expires_in": 86400,
  "token_type": "Bearer"
}
```

1.4 Get Access Token and ID using the Refresh Token

Using any API testing tool, submit a POST request to

<https://account.uipath.com/oauth/token> with the body below, replacing the values for **[refresh_token]** accordingly.

```
{
  "grant_type": "refresh_token",
  "client_id": "5v7PmPJL6FOGu6RB8I1Y4adLBhIwovQN",
  "refresh_token": "[refresh_token]"
}
```

Copy the values for **access_token** and **id_token** from the response.

```
{
  "access_token": "eyJ0eX...",
  "id_token": "eyJ0eX...",
  "scope": "openid profile email offline_access",
  "expires_in": 86400,
  "token_type": "Bearer"
}
```

2. Get UiPath Account, Service and Process Identifiers

2.1 Get the Account Logical Name

Using any API testing tool, submit a GET request to

<https://platform.uipath.com/cloudrpa/api/getAccountsForUser> with a request header

```
Authorization: Bearer [id_token]
```

Copy the value for **accountLogicalName** from the response:

```
{
  "userEmail": "demo@domain.com",
  "accounts": [
    {
      "accountName": "Demo",
      "accountLogicalName": "Demo"
    }
  ]
}
```

2.2 Get Service Instance Logical Name

Using any API testing tool, submit a GET request to

[https://platform.uipath.com/cloudrpa/api/account/\[accountLogicalName\]/getAllServiceInstances](https://platform.uipath.com/cloudrpa/api/account/[accountLogicalName]/getAllServiceInstances) with a request header

```
Authorization: Bearer [id_token]
```

Copy the value for **serviceInstanceLogicalName** from the response:

```
[
  {
    "serviceInstanceName": "DemoDefault",
    "serviceInstanceLogicalName": "DemoDefaultzous50676",
    "serviceType": "ORCHESTRATOR",
    "serviceUrl": "https://platform-community.azurewebsites.net"
  }
]
```

2.3 Get Release Key for the Desired Process

We will now need to retrieve the Release Key, a unique identifier for a desired process. Using any API testing tool, submit a GET request to

<https://platform.uipath.com/odata/Releases> with request headers

```
Authorization: Bearer [access_token]
X-UIPATH-TenantName: [serviceInstanceLogicalName]
```

Copy the value for **Key** from the response:

```
{
"@odata.context": "https://platform.uipath.com/odata/$metadata#Releases",
"@odata.count": 1,
"value": [
  {
    "Key": "b27c7363-459c-4520-bae5-660d4ald3813",
    "ProcessKey": "Demo_Process",
    "ProcessVersion": "1.0.21",
    "IsLatestVersion": false,
    "IsProcessDeleted": false,
    "Description": "Demo Process",
    "Name": "Demo Process",
    "EnvironmentId": 98069,
    "EnvironmentName": "Demo Environment",
    "InputArguments": null,
    "QueueDefinitionId": null,
    "QueueDefinitionName": null,
    "Id": 120897,
    "Arguments": {
      "Input": null,
      "Output": null
    },
    "ProcessSettings": null
  }
]
}
```

2.4 Test Orchestrator API Call to Start a Job

At this point, make sure you have the following critical information at hand:

- refresh_token (Refresh Token)
- access_token (Access Token)
- serviceInstanceLogicalName (Service Instance Logical Name)
- Key (Release Key)

Let's try making an API call to start a job for the process.

Using any API testing tool, submit a POST request to

<https://platform.uipath.com/odata/Jobs/UIPath.Server.Configuration.OData.StartJobs> with 2 request headers

```
Authorization: Bearer [access_token]
X-UIPATH-TenantName: [serviceInstanceLogicalName]
```

and the body:

```

    { "startInfo":
    { "ReleaseKey": "[Key]",
      "Strategy": "All",
      "RobotIds": [ ],
      "JobsCount": 0,
      "Source": "Manual"
    }
  }
}

```

If successful, the response will be as follows, with the status of the Job shown in the State attribute:

```

{
"@odata.context": "https://platform.uipath.com/odata/$metadata#Jobs",
"value": [
  {
    "Key": "e811bd29-26fb-4cc3-af91-7fd6308ca643",
    "StartTime": null,
    "EndTime": null,
    "State": "Pending",
    "Source": "Manual",
    "SourceType": "Manual",
    "BatchExecutionKey": "382e7a96-a5b9-4343-b258-26a80cf87f80",
    "Info": null,
    "CreationTime": "2019-06-12T08:02:03.2595408Z",
    "StartingScheduleId": null,
    "ReleaseName": "Demo Process",
    "Type": "Unattended",
    "InputArguments": null,
    "OutputArguments": null,
    "HostMachineName": null,
    "HasMediaRecorded": false,
    "Id": 10169582
  }
]
}

```

Integrate UiPath with Joget

How to Integrate with UiPath

UiPath provides the [Orchestrator API](#) for integration.

The most commonly used function is [Starting a Job](#), which should satisfy a majority of use cases. We will use this function in the following integration tutorial.

Design a Process to Start a UiPath Job

Since the UiPath Orchestrator API is a [REST](#) API with data in [JSON](#) format, we can use the Joget [JSON Tool](#) to invoke the API.

Before we start, ensure that you have the following critical UiPath information at hand (obtained from the previous Prepare for UiPath Integration tutorial):

- refresh_token (Refresh Token)
- serviceInstanceLogicalName (Service Instance Logical Name)
- Key (Release Key)

Step 1. Design New App

First, let's design a new app by clicking on Design New App in the Joget [App Center](#).

[blocked URL](#)

Fill in desired values for the App ID and App Name, and click Save.

Step 2. Design New Process

Click on the Processes menu, then on the Design Process button to launch the [Process Builder](#).

Design a simple process containing 2 tools, as shown below.

[blocked URL](#)

Click on the edit pencil icon after hovering over the process name along the top. Enter a suitable process name and create 3 workflow variables to store the response values from UiPath API call:

- status
- id_token
- access_token

[blocked URL](#)

Click on Deploy.

Step 3. Configure First Tool to Get UiPath Access Token

Once the process has been deployed, close the Process Builder. In the process view page, select the Map Tools to Plugins tab.

[blocked URL](#)

Click on Configure Plugin for the first tool and select JSON Tool, then key in the following configuration:

JSON URL	https://account.uipath.com/oauth/token
Call Type	POST
POST Method	Custom JSON Payload
Custom JSON Payload	<pre>{ "grant_type": "refresh_token", "client_id": "5v7PmPJL6FOGu6RB8l1Y4adLBhIwovQN", "refresh_token": "[refresh_token]" }</pre>

[blocked URL](#)

Under Store to Workflow Variable, map the variables to store the tokens in the matching workflow variables i.e.

access_token	access_token
id_token	id_token

[blocked URL](#)

Click on Submit to save.

Step 4. Configure Second Tool to Start Job Using the Access Token

Click on Configure Plugin for the second tool and select JSON Tool, then key in the following configuration:

JSON URL	https://platform.uipath.com/odata/jobs/UiPath.Server.Configuration.OData.StartJobs
Call Type	POST
POST Method	Custom JSON Payload
Custom JSON Payload (Replace [Key] with the actual Release Key)	<pre>{ "startInfo": { "ReleaseKey": "[Key]", "Strategy": "All", "RobotIds": [], "JobsCount": 0, "Source": "Manual" } }</pre>
Request Headers (Replace [serviceInstanceLogicalName] with the actual Service Instance Logical Name)	Authorization: Bearer #variable.access_token# X-UIPATH-TenantName: [serviceInstanceLogicalName]

[blocked URL](#)

Under Store to Workflow Variable, map the status variable to the State attribute in the response JSON i.e.

status	value[0].State
--------	----------------

[blocked URL](#)

That's it. The Joget process has been configured to invoke the UiPath Orchestrator API to start a job.

Step 5. Run the Process

Now, let's test the process. Click on the Run Process button at the top, then Run Process again in the confirmation dialog.

[blocked URL](#)

Once the process has started, the 2 tools will execute as configured. To view the results of the process, navigate to Monitor -> Completed Processes.

Select the process instance and you will see the 2 tools executed.

[blocked URL](#)

Click on each activity to view the values of the workflow variables which obtained the results of the Orchestrator API calls.

[blocked URL](#)

If you are running Joget on-premise and have access to the system logs, you will be able to see the tool requests and responses if the debugging option is enabled.

Sample log output:

```
INFO 12 Jun 2019 09:28:38 org.joget.apps.app.lib.JsonTool - POST : https://account.uipath.com/oauth/token
INFO 12 Jun 2019 09:28:38 org.joget.apps.app.lib.JsonTool - Custom JSON Payload : {_ "grant_type":
"refresh_token",_ "client_id": "5v7PmPJL6FOGu6RB8I1Y4adLBhIwovQN",_ "refresh_token":
"CBZcOD6vrP2FQ9qa8fuqDdfoEwnVPuR2Kpz-hitbTAIzG"}_
INFO 12 Jun 2019 09:28:39 org.joget.apps.app.lib.JsonTool - https://account.uipath.com/oauth/token returned
with status : 200
INFO 12 Jun 2019 09:28:39 org.joget.apps.app.lib.JsonTool - {"access_token":"eyJ0eX...", "scope":"openid profile
email offline_access", "expires_in":86400, "token_type":"Bearer"}
INFO 12 Jun 2019 09:28:39 org.joget.apps.app.lib.JsonTool - POST : https://platform.uipath.com/odata/Jobs
/UiPath.Server.Configuration.OData.StartJobs
INFO 12 Jun 2019 09:28:39 org.joget.apps.app.lib.JsonTool - Custom JSON Payload : { "startInfo":_ {
"ReleaseKey": "b27c7363-459c-4520-bae5-660d4ald3813",_ "Strategy": "All",_ "RobotIds": [ ],_ "JobsCount": 0,_
"Source": "Manual" _ } _}_
INFO 12 Jun 2019 09:28:39 org.joget.apps.app.lib.JsonTool - Adding request header Authorization : Bearer
eyJ0eX...
INFO 12 Jun 2019 09:28:39 org.joget.apps.app.lib.JsonTool - Adding request header X-UIPATH-TenantName :
DemoDefaultzous50676
INFO 12 Jun 2019 09:28:40 org.joget.apps.app.lib.JsonTool - https://platform.uipath.com/odata/Jobs/UiPath.
Server.Configuration.OData.StartJobs returned with status : 201
INFO 12 Jun 2019 09:28:40 org.joget.apps.app.lib.JsonTool - {"@odata.context":"https://platform.uipath.com/odata
/$metadata#Jobs", "value":[{"Key":"aa667a3e-04f9-4013-a8eb-ce701dac8799", "StartTime":null, "EndTime":null, "
State":"Pending", "Source":"Manual", "SourceType":"Manual", "BatchExecutionKey":"68fa4d0a-0fcc-4d49-ae0d-
b6631a7dfc78", "Info":null, "CreationTime":"2019-06-12T09:28:12.9207318Z", "StartingScheduleId":null, "
ReleaseName":"Demo Process", "Type":"Unattended", "InputArguments":null, "OutputArguments":null, "HostMachineName":
null, "HasMediaRecorded":false, "Id":10173064}]}
```

If you are using [Joget DX](#), you can stream the logs directly in the web browser.

[blocked URL](#)

Back at the UiPath Orchestrator and UiPath Robot, you would also be able to monitor the execution of the job.

Sample Joget App

The sample app containing this process can be downloaded below:



APP_rpa_uipath...0612141440.jwa