INTEGRATION MODULE FOR SALESFORCE

For Kelverion Runbook Studio and Azure Automation

User Guide

Version 1.1

Kelverion Integration Module for Salesforce

Published: March 2021

The Kelverion Integration Module for Salesforce is Microsoft Azure Certified

Feedback

Send suggestions and comments about this document to support@kelverion.com

Contents

Getting Started

The following sections outline how to deploy and configure the Kelverion Integration Module for Salesforce.

System Requirements

The Integration Module for Salesforce requires the following software to be installed and configured prior to implementing the integration. For more information about installing and configuring Salesforce, refer to the respective product documentation.

- Kelverion Runbook Studio 4.2
- Microsoft .NET Framework 4.7.2
- Salesforce Force.com based application with Salesforce API access

Important: The Integration Module for Salesforce requires Salesforce API access, which is not available by to all Salesforce editions. Programmatic access is available to the Enterprise, Unlimited, Developer and Performance editions. If you are using the Professional edition, API access must be purchased and can be arranged by contacting your Salesforce account executive.

Important: The integration module uses the Force.com REST API with OAuth 2.0 authentication. You must set up a Salesforce **connected app** as part of your environment. For details, see **Connecting to Salesforce** in this user guide.

Installing the Integration Module

The easiest way to install and deploy the Integration Module for Salesforce is from the PowerShell Gallery, but you can also download the module from Kelverion and perform the steps manually.

You must install and deploy the integration module to each Azure Automation account and Hybrid Worker host system that you plan to use to run your runbooks. You must also install the integration module on any Runbook Studio host systems that you will be using to build and manage your runbooks.

Using the PowerShell Gallery

Using the commands in the **PowerShellGet** module you can download the Integration Module for Salesforce from the PowerShell Gallery and install it on your local computer. You can also deploy the module directly from the PowerShell Gallery to any of your Azure Automation Accounts.

Install the Integration Module on your local computer or Hybrid Worker:

1. Confirm that the latest **PowerShellGet** module is installed.
2. Start a PowerShell window as an Administrator and run the command:

   ```
   Install-Module -Name Kelverion.Salesforce -Scope AllUsers
   ```

Upload the Integration Module to an Azure Automation account:

1. Go to the PowerShell Gallery.
2. Click the **Azure Automation** tab.
3. Click **Deploy to Azure Automation**. You will be directed to Microsoft Azure.
4. Select the **Automation Account** that you want to deploy the module to.
5. Click **OK**.

Manual Installation

Alternatively, you can download the integration module package from Kelverion and deploy it manually to your local computer, hybrid workers and Azure Automation accounts.

The download package from Kelverion includes a **.zip** file containing the integration module as well as the User Guide and Release Notes. The following instructions assume that you have unzipped the download package and have access to the **.zip** file containing the integration module.

Install the Integration Module on your local computer or Hybrid Worker:

1. Copy the **Kelverion.Salesforcel.zip** file to your local computer.
2. Right-click on the file and select **Properties**.
3. Click the **General** tab. If necessary, click **Unblock**, and then click **OK**.
4. Unzip the **Kelverion.Salesforcel.zip** file.
5. Copy the **Kelverion.Salesforce** folder to a location in the `%PsModulePath%` path.

Important: When installing the integration module on a hybrid worker, you must use a location that is accessible to all users of the computer.

Upload the Integration Module to an Azure Automation account:

1. Sign into Microsoft Azure.
2. Open the Automation Account that you want to upload the module to.
3. Click **Modules** under Shared Resources. The list of installed modules is displayed.
4. Click **Add a module** at the top of the list.
5. In the **Upload File** box, select the **Kelverion.Salesforce.zip** file that you downloaded.
6. Click **OK**. Importing the module may take several minutes.

Licensing the Integration Module

Licenses for Kelverion integration modules are managed and deployed using the *Kelverion Runbook Studio* and Azure *Automation connection assets*.

Register an Integration Module license with Runbook Studio:

1. Open **Kelverion Runbook Studio**.
2. On the **File** tab, click **About**.
3. Click **License Information**.
4. Click the **Integration Modules** tab, and then click **Add License.**
5. Select the integration module license file (.kaml) and click **Open**.
6. You should see your entitlements displayed in the list.
7. Click **OK**.

Important: Entitlements will not display until after the integration module has been installed on the Runbook Studio computer.

Create a Connection Asset with a license key and upload it to Azure:

1. On the **Home** tab, click **Sign In**. The Sign In dialog appears.
2. Sign into your account.
3. In the **Active Azure Automation Account** box, select the account that you want to add the connection asset to.
4. Click **New Asset** and then click **Connection**. The New Connection dialog appears.
5. In the **Name** field, enter a name to identify the connection.
6. In the **Connection Type** field, select the desired connection type.
7. Enter the appropriate connection information in the provided fields.

8. Click **OK**.

Update all Connection Asset license keys and upload them to Azure:

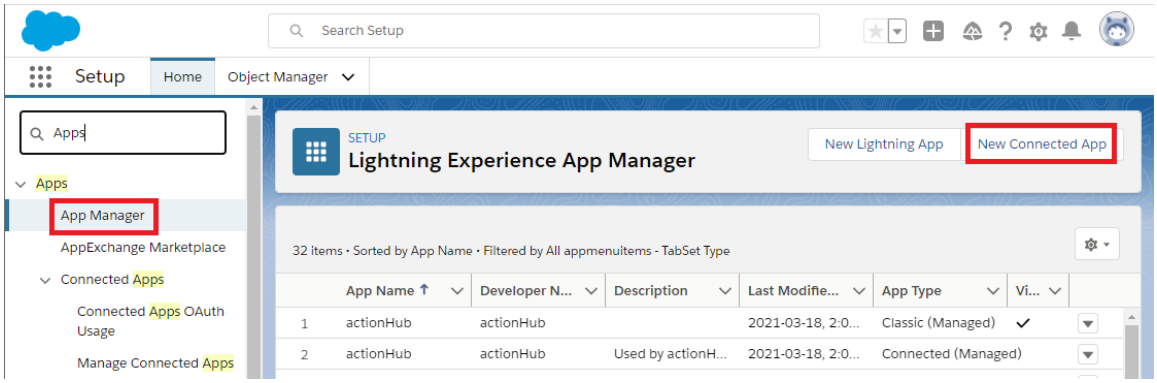
1. On the **Home** tab, click **Sign In**. The Sign In dialog appears.
2. Sign into your account.
3. In the Explorer panel, click the **Azure (Online)** group.
4. Right-click the Azure Automation Account that contains the connection assets you want to update, and then click **Update License Keys**. A summary is displayed.

Connecting to Salesforce

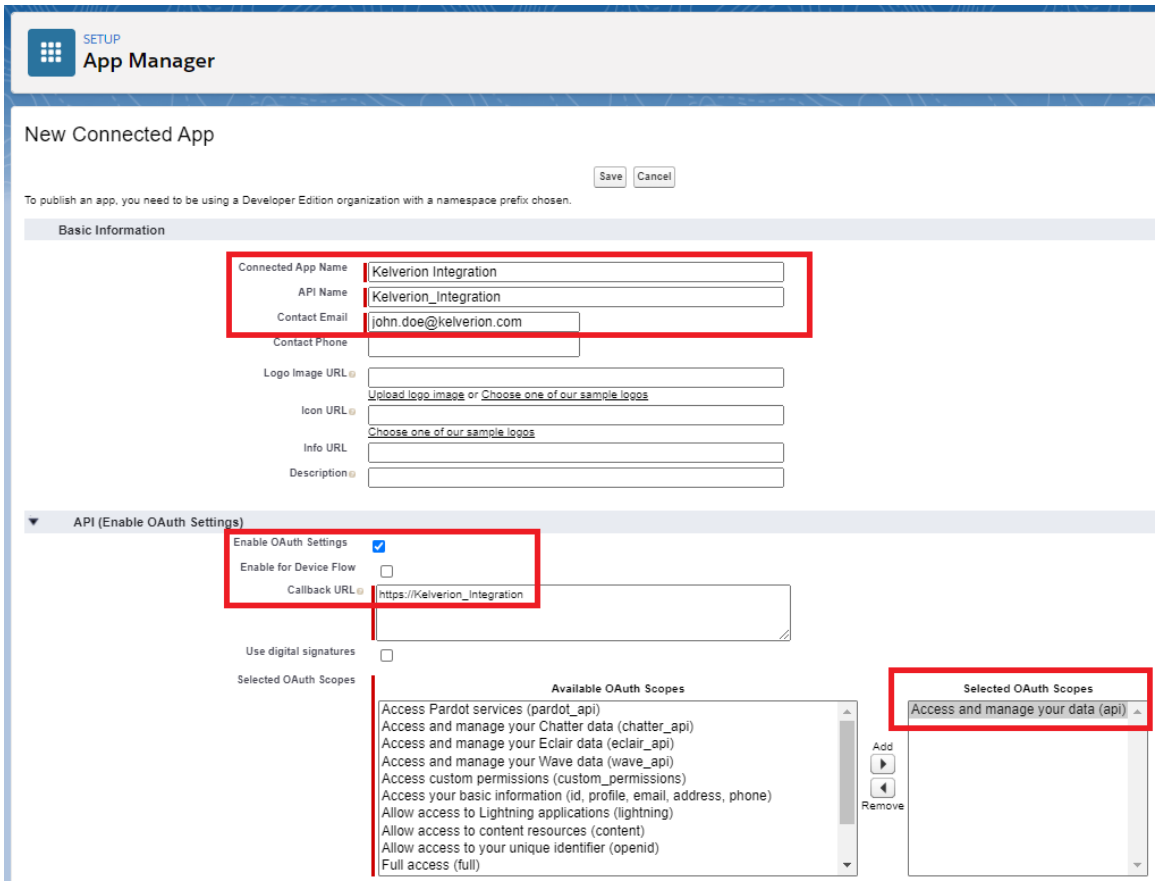
Salesforce Force.COM REST API uses the OAuth protocol to allow users to securely access their Salesforce environment. For the Integration Pack to gain access to your Salesforce environment, in your Salesforce organization, you must configure a connected app that defines OAuth settings. For more information, see Create a Connected App in the online documentation for Salesforce.

Use the following steps to create a new connected app. The user performing this should be configured with **System Administrator** profile.

1. Click on Setup on upper right side under the gear menu, and then enter “Apps” in the Quick Find box.
2. Select **App Manager** under Apps.
3. In the **App Manager** click **New Connected App** to create a new connected app.



4. Enter the **Name** of your application, for example *KelverionSalesforceModule*.
5. Enter **Contact Email** information, as well as any other information appropriate for your application.
6. Select **Enable OAuth Settings**.
7. The **Callback URL** is required; however, this is not needed for integrating with the integration pack. You can enter an arbitrary HTTPS URL, for example https://Kelverion_Integration.
8. In the **Selected OAuth Scopes** add **Access and manage your data (api)**.



9. Click **Save** and then **Continue**.
10. After the connected app is created, copy the **Consumer Key** and the **Consumer Secret** and enter them in their respective boxes in IP configuration.

Salesforce API Limits

The Kelverion Integration Module for Salesforce uses Salesforce REST API to integrate with your Salesforce environment. Salesforce imposes several limits on its REST API and the IP is also subject to these limitations and restrictions. For more information, see API Request Limits and Allocations and API calls reports and limits FAQ in the online documentation for Salesforce.

Daily API Calls (Requests) Limit

The main API limit affecting the Integration Pack is the **Daily API Calls (or Requests) limit**. This value controls how many requests IP activities can send to Salesforce application, per day.

The maximum number of daily API calls depends on your Salesforce organization and edition. Once the maximum number of daily API calls has been reached, IP activities will fail with REQUEST_LIMIT_EXCEEDED exception.

The Get Limits Activity

To help control the requests volume of your Salesforce integration, the IP provides the **Get Limits** activity, which can be used to check the status of your daily limits.

Before running the integration pack in a production environment, it is recommended to measure/estimate your typical daily API request volume. If your integration needs tend to reach close to the daily limits for your organization, the Get Limits activity can be used to throttle your Salesforce runbooks such that the API call limit is not reached during periods of high volume.

Concurrent API Calls (Requests) Limit

Another important limit to keep in mind is the Concurrent API Requests limit. This value specifies the maximum number of concurrent calls that the integration pack can make, with a duration of 20 seconds or longer.

To avoid reaching this limit, it is recommended to design Salesforce activities in your workflow sequentially.

Stored Salesforce Information

The integration module uses Salesforce object metadata to provide object specific parameters, filters, and outputs. In order to improve performance and usability in Runbook Studio, the IM stores object and field information locally on the Runbook Studio machine. All information stored by the IM can be found at the following location:

[Program Data Directory]\Kelverion\Salesforce\Schema_v1.0\[Salesforce Server]\Schema

This folder contains object and field information for salesforce objects configured in the IM. Stored data grows gradually as you use more Salesforce object types.

Initially the IM stores the *Schema*/*SObjects.json* file, which contains summary information about all Salesforce objects available in your organization. Afterwards, for each object type accessed by the IM, detailed information is stored in a corresponding JSON file. For example, *Contact* object information is stored in **Schema/*Contact.json***

If a new custom object is added in your organization, the information in the *SObjects.json* must be refreshed. The best way to accomplish this is to first delete the *SObjects.json* file, and then refresh one of the IM activities. This will result in the IM saving a new up-to-date *SObjects.json* metadata file.

Similarly, if the definition of an existing object changes in your organization, for example if a field is added or removed, the stored object schema information must be refreshed. The best way to accomplish this is to identify the corresponding JSON file in the *Schema* folder, delete it, and then refresh an IM activity of that type. This will result in the IM saving a new up-to-date object schema file.

Working with Activities in Runbook Studio

The following sections outline some of the common configuration options that are available to you when working with the smart activities in the Kelverion Integration Module for Salesforce in Kelverion Runbook Studio.

The advanced discovery capabilities provided by the activities in this integration module are only supported when authoring runbooks in Kelverion Runbook Studio.

When you publish your runbooks to Azure Automation, Runbook Studio will automatically convert the dynamically generated parameters and filters of your smart activities into fully configured Azure Automation graphical runbook activities. Similarly, when generating PowerShell snippets, Runbook Studio will convert the dynamically generated parameters and filters in your smart activities into the appropriate PowerShell expression.

The integration module includes the following activities:

Get-SalesforceLimits	Retrieves Salesforce API limits
Get-SalesforceRecord	Retrieves and filters Salesforce records
New-SalesforceRecord	Creates a new record
Remove-SalesforceRecord	Remove an existing record
Set-SalesforceRecord	Update an existing record

Smart Connections

In Kelverion Runbook Studio you can configure one or more Smart Connections to establish reusable links between Runbook Studio and specific EasyVista instances. You can create as many Smart Connections as required.

Adding a Smart Connection to Kelverion Runbook Studio:

1. On the **Home** tab, click **Smart Connections**. The Smart Connections dialog appears.
2. Click **Add a connection** at the top of the list.
3. In the **Name** box, enter a name for the connection.
4. In the **Connection Type** box, select *Kelverion.Salesforce*.
5. In the **ServerUrl** box, enter the URL to your Salesforce site. For example: *https://na22.salesforce.com.*
6. In the **LoginUrl** box, enter the URL to the Salesforce login service. Typically, this is *https://login.salesforce.com/services/oauth2/token*.
7. In the **Username** and **Password** boxes, enter the credentials used to administer your Salesforce environment.

8. In the **SecurityToken** box, enter the security token for the user specified in the previous step. See Obtaining User Security Token for more information.
9. In the **ClientId** box, enter the consumer key for the connected app which you are using to integrate with your Salesforce environment. See Connecting to Salesforce for more information.
10. In the **ClientSecret** box, enter the consumer secret for the connected app which you are using to integrate with your Salesforce environment. See Connecting to Salesforce for more information.
11. In the **SessionTimeoutMinutes**, enter the Salesforce Session Timeout value, in minutes. See below for more information.
12. Add additional connections if applicable.
13. Click **OK** to close the configuration dialog box, and then click **OK**.

Tip: This **SessionTimeoutMinutes** value must be less than or equal to the Session Timeout as configured in Session Settings on your Salesforce environment. You can find your Session Timeout in your Salesforce portal at **Setup > Administer > Security Controls > Session Settings**.

If the value configured in your Session Settings is in hours, you must convert this into minutes and then specify it in Session Timeout (Minutes) box. By default, this value is set to 15 minutes, which is the lowest available setting.

Azure Automation Connection Assets

The activities in the Kelverion Integration Module for Salesforce require connection information to connect to instances of Salesforce from Azure Automation. The recommended way to pass connection information to your activities is to use Azure Automation connection assets.

Connection assets let you securely define connection information in Azure which can then be retrieved on demand using either the **Get-AutomationConnection** cmdlet or connection asset data source.

Add an Azure Automation connection asset in Runbook Studio:

1. On the **Home** tab, click **Sign In**. The Sign In dialog appears.
2. Sign into your account.
3. In the **Active Azure Automation Account** box, select the account that you want to add the connection asset to.
4. Click **New Asset** and then click **Connection**. The New Connection dialog appears.
5. In the **Name** box, enter a name for the configuration.
6. In the **Connection Type** box, select **Kelverion.Salesforce**.
7. In the **ServerUrl** box, enter the URL to your Salesforce site. For example: https://20210319010417039 5.my.salesforce.com. Note that the lightning URL displayed in

your web browser may not work, instead use the site URL as configured under Setup > User Interface > Sites and Domains > Domains.

8. In the **LoginUrl** box, enter the URL to the Salesforce login service. Typically, this is https://login.salesforce.com/services/oauth2/token.
9. In the **Username** and **Password** boxes, enter the credentials used to administer your Salesforce environment.
10. In the **ClientId** box, enter the consumer key for the connected app which you are using to integrate with your Salesforce environment. See Connecting to Salesforce for more information.
11. In the **ClientSecret** box, enter the consumer secret for the connected app which you are using to integrate with your Salesforce environment. See Connecting to Salesforce for more information.

    In the **SessionTimeoutMinutes**, enter the Salesforce Session Timeout value, in minutes, for your organization. See below for more information.
12. Click **OK** to close the New Connection dialog box.

Tip: This **SessionTimeoutMinutes** value must be less than or equal to the Session Timeout as configured in Session Settings on your Salesforce environment. You can find your Session Timeout in your Salesforce portal at **Setup > Administer > Security Controls > Session Settings**.

If the value configured in your Session Settings is in hours, you must convert this into minutes and then specify it in Session Timeout (Minutes) box. By default, this value is set in the integration module to 15 minutes, which is the lowest available setting.

Activity Properties

All activities in the Kelverion Integration Module for Salesforce have the following properties:

Label	A unique label that identifies the activity in the runbook. Runbook Studio will provide a default name for each activity, but you can provide your own labels to make their role in the runbook more obvious.
Description	An optional description of the activity. Providing a description is a great way to let everyone understand the function of the activity in the runbook.
Checkpoint	Indicates whether or not a checkpoint is set in the runbook workflow after the activity runs. Checkpoints are only available for Graphical PowerShell Workflow runbooks. If the runbook uses Azure cmdlets you should follow best practices and follow a check-pointed activity with an **Add-AzureRMAccount** in case the runbook is suspended and restarts from this checkpoint on a different worker.

Smart Discovery

When designing runbooks in Kelverion Runbook Studio, you will notice that the activities in the Kelverion Integration Module for Salesforce include a **Discovery** panel instead of the **Parameter Sets** panel that is present for standard command activities. This is because the activities in the Kelverion Integration Module for Salesforce support interactive discovery of server assets in your environment.

All activities in the Kelverion Integration Module for Salesforce have a **Connection** option on the **Discovery** panel, which lets you specify a smart connection, so that Runbook Studio can connect to Salesforce.

When connected to Salesforce, Runbook Studio will provide a list of Salesforce objects which allow you to specify what type of object you want to Create/Retrieve/Update/Delete. Once you have selected an object type, Runbook Studio will provide additional parameters and filters specific for that object.

Smart Parameters

Unlike standard command activities whose parameters are determined by the Parameter Set that is selected, the parameters in the Kelverion Integration Module for Salesforce are determined by the Discovery options that you specify.

For example, when using the **New-SalesforceRecord** activity, the Discovery panel will contain options for selecting the Salesforce object type. If you select the **Account** object type, for example, Runbook studio will provide parameters that are specific for the Account object.

You must configure all mandatory parameters. To view the optional parameters that are associated with an activity, click **Optional** at the top of the Parameters tab.

All activities in the Kelverion Integration Module for Salesforce include a **Connection** parameter. This should not be confused with the similarly named Connection property on the Discovery panel. The former specifies runtime connection information, which the activity uses when it executes as part of a runbook, in Azure or on the Hybrid Worker. The later specifies design-time connection information, so that Runbook Studio can access Salesforce and provide object specific design-time configuration options.

Several factors determine the data sources that are available when configuring a parameter. They include: the parameter's data type, whether it is linked to another activity and whether the runbook has any input parameters.

Runbook studio supports the following data sources.

Activity output	Specify activity whose output will be assigned to the parameter. You may also provide an optional Path to select a specific property of the output objects that are generated by the activity. Available when the activity is linked to a source activity.
Not configured	Clears any value that was previously configured. You must configure all mandatory parameters.

Certificate asset	Specify the name of the global certificate asset that will be used to provide a value for the parameter. If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the certificates that are available.
Credential asset	Specify the name of the global credential asset that will be used to provide a value for the parameter. If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the credentials that are available.
Constant	Specify a constant value to assign to the parameter. Available for parameters that have the following data types: • String • DateTime • Boolean • Char • Byte • SByte • Int16 • Int32 • Int64 • UInt16 • UInt32 • Uint64 • Decimal • Double • Float • SwitchParameter When assigning a constant DateTime value, Runbook Studio assumes the value is in UTC.
Connection asset	Specify the name of the global connection asset that will be used to provide a value for the parameter. If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the connections that are available.
Empty string	An empty string will be assigned to the parameter. Available when the parameter is type *System.String*
Null	A null ($null) value will be assigned to the parameter. Available when the parameter type is a reference type.
PowerShell expression	Specify a *simple* PowerShell expression whose output will be assigned to the parameter. You can use variables in the expression to access the output of an activity or a runbook parameter.
Runbook input	Specify the name of the runbook input parameter whose value will be assigned to the parameter. Available when the runbook has one or more input parameters.
Variable asset	Specify the name of the global variable asset that will be used to provide a value for the parameter. If you have connected to Azure and selected a Subscription and Automation Account on the toolbar, the data source will provide the names of the variables that are available.

Smart Filters

Some of the activities in the Kelverion Integration Module for Salesforce include a Filters panel which lets you specify filters that can be used to retrieve a sub-set of Salesforce records.

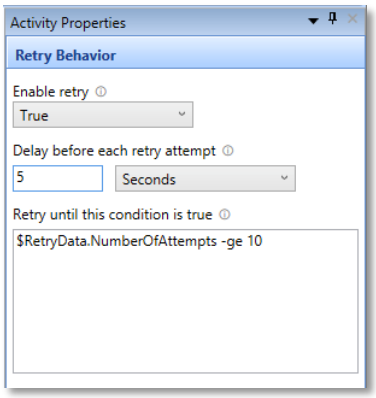
If you do not provide any filters, then the activity will retrieve all objects for that activity.

To add a filter to your activity, select the **Filters** panel and click **Add.** Filters have the following properties.

Filter	The name of the filter.
Operation	The operation used to evaluate the filter. Different operators will be provided based on the filter that is selected. Possible filter operators are: • Equals • Does not equal • Contains • Does not contain • Starts with • Ends with • Matches • Does not match • Greater than • Greater than or equal to • Less than • Less than or equal to
Value	The data source used to retrieve the value used to evaluate the filter. The value used to evaluate the filter will be obtained. For more information on data sources, please refer to the Parameters section for more information on configuring data sources.

Retry Behaviour

The activities in the Kelverion Integration Module for Salesforce can be configured to run multiple times until a particular condition, which you specify, is satisfied. You can use the retry behavior options to configure activities that should run multiple times, that are error prone or may need more than one attempt for success.



When you enable retry for an activity, you can configure the runbook to wait a specified number of minutes or seconds before running the activity again. If no delay is specified the runbook will run the activity again, immediately after it completes.

The retry condition lets you specify a PowerShell expression that the runbook will evaluate after each time the activity runs. If the result of the expression is true the activity does not run again, and the runbook moves on to the next child activity in the runbook.

When defining the retry conditions for your activity you can take advantage of a global variable called **$RetryData**. Specific information about the last time the activity ran can be accessed using the following properties.

Property	Description
NumberOfAttempts	Number of times that the activity has ran
Output	Output that was generated by the activity the last time that it ran
TotalDuration	Time elapsed since the activity was started
StartedAt	Time in UTC when the activity was first started

The following are some examples of activity retry conditions

```
# Run the activity exactly 5 times
$RetryData.NumberOfAttempts -eq 5

# Run the activity until it produces some output
$RetryData.Output.Count -ge 1

# Run the activity until at least 2 minutes has elapsed
$RetryData.TotalDuration.TotalMinutes -ge 2
```

Additional Parameters

The activities in the Kelverion Integration Module for Salesforce let you specify additional PowerShell parameters that you can use to control the behavior of the activity.

For example, to output detailed information about the operation performed by an activity you would specify **-Verbose:$True**

Get-SalesforceLimits

The **Get-SalesforceLimits** activity can be used in a runbook to retrieve Salesforce API limits for your Salesforce environment. The activity returns both the maximum allowed, and the current (remaining) values for the day.

You can use this activity to monitor the status of your daily API calls (requests). You may need to redesign your runbooks or limit how often they run if you frequently surpass Salesforce's API usage limits. See Salesforce API Limits for more information.

Tip: Running this activity counts towards your Daily API Requests limit and we recommend that it is used sparingly.

Discovery Options

Use the following options to connect to Salesforce and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Salesforce.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.

Outputs

This activity outputs an object with the following properties.

DailyApiRequests Max	The maximum number of Daily API requests (calls) allowed for your organization.
DailyApiRequests Remaining	The remaining number of API requests (calls) for the day.
DailyAsyncApex ExecutionsMax	The maximum number of daily async Apex executions allowed for your organization.
DailyAsyncApex ExecutionsRemaining	The remaining number of async Apex executions for the day.
DailyBulkAPI RequestsMax	The maximum number of daily Bulk API requests (calls) allowed for your organization.
DailyBulkAPI RequestsRemaining	The remaining number of Bulk API requests (calls) for the day.
DataStorageMbMax	Maximum data storage for your organization.
Data StorageMb Remaining	Remaining data storage for your organization.

Get-SalesforceRecord

The **Get-SalesforceRecord** activity can be used in a runbook to retrieve records of the specified object type.

Discovery Options

Use the following options to connect to Salesforce and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Salesforce.
Object Type	Specifies the type of Salesforce objects to retrieve. Note that the selected object type includes the internal name in square brackets. For example, *Account [Account]*
Publish All Fields	Specifies whether the activity will return all record fields, or just a subset of *summary* fields. Summary fields typically include record identifier fields, name fields, Created Date and Last Modified Date.
Search By	Specifies how Salesforce records are to be retrieved. • **Filters**. You can specify one or more filters to narrow down the result set. • **ID**. You can specify a record ID to retrieve a specific record.

Tip: Choosing not to publish all record fields can improve performance in cases where specific details for an object are not needed, for example, when only the Record ID for a specific record is required. Regardless of whether you chose to publish all record fields or not, the activity will provide all available filters for the specified object.

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Record Limit	The maximum number of records the activity should retrieve.

Optional Parameters

You can configure the following parameters, as required.

Descending	When **OrderBy** is specified, this property specifies whether the returned records will be ordered in ascending or descending order. When not specified, the activity returns records in ascending order.
OrderBy	Specifies a field used for ordering the returned records. When not specified, there is no guarantee of the records order.

Filters

This activity provides filters, based on the **Object Type** that was selected. You can combine filters to selectively control which object records to retrieve.

Outputs

This activity outputs objects that represent the records that were retrieved from Salesforce. The properties of these objects are based on the **Object Type** that was selected.

New-SalesforceRecord

The **New-SalesforceRecord** activity can be used in a runbook to insert a new record of the specified object type.

Discovery Options

Use the following options to connect to Salesforce and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Salesforce.
Object Type	Specifies the type of Salesforce object to create. Note that the selected object type includes the internal name in square brackets. For example, *Account [Account]*

Required Parameters

You must configure the following parameters. Additional required parameters may be provided, depending on the **Object Type** that was selected.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.

Optional Parameters

This activity may provide optional parameters, depending on the **Object Type** that was selected, and you can configure them as required.

Outputs

The activity outputs the **Record ID** of the Salesforce object that was created.

Remove-SalesforceRecord

The **Remove-SalesforceRecord** activity can be used in a runbook to delete a record of the specified object type.

Discovery Options

Use the following options to connect to Salesforce and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Salesforce.
Object Type	Specifies the type of Salesforce object to remove. Note that the selected object type includes the internal name in square brackets. For example, *Account [Account]*

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Record ID	Identifies the record to be deleted.

Outputs

The activity outputs the **Record ID** of the Salesforce record that was removed. This Record ID is no longer valid and should not be used by downstream activities.

Set-SalesforceRecord

The **Set-SalesforceRecord** activity can be used in a runbook to update an existing Salesforce record of the specified object type.

Discovery Options

Use the following options to connect to Salesforce and configure the activity.

Connection	The name of the Smart Connection used to connect Runbook Studio to Salesforce.
Object Type	Specifies the type of Salesforce object to update. Note that the selected object type includes the internal name in square brackets. For example, *[Account]*

Required Parameters

You must configure the following parameters.

Connection	A hashtable containing connection information. This is typically obtained using a Connection Asset data source or Get-AutomationConnection activity.
Record ID	Identifies the record to be updated.

Optional Parameters

This activity may provide optional parameters, depending on the **Object Type** that was selected, and you can configure them as required.

Outputs

The activity outputs the **Record ID** of the Salesforce object that was updated.