



# **AUTOMATION PORTAL**

**User Guide**

Version 3.8

# Kelverion Automation Portal

Copyright 2017 Kelverion Inc. All rights reserved.

Published: September 2023

## ***Feedback***

Send suggestions and comments about this document to [support@kelverion.com](mailto:support@kelverion.com)

# Table of Contents

Table of Contents .....	3
Trademarks .....	7
Using this Guide .....	7
Accessibility Statement.....	7
Installing and Administrating the Automation Portal.....	8
System Requirements.....	8
<i>On-Prem</i> .....	8
<i>Azure</i> .....	8
<i>Web Browser</i> .....	9
<i>Backup</i> .....	9
Installing On-Prem.....	9
<i>Internet Information Services (IIS) settings</i> .....	11
<i>Enable Always Encrypted</i> .....	11
<i>Always Encrypted Certificate Import/Export</i> .....	13
<i>High Availability (HA) Deployment</i> .....	16
<i>Additional Configuration</i> .....	16
Upgrading On-Prem.....	17
Installing in Azure .....	18
<i>Locating Azure Ids</i> .....	18
<i>PowerShell</i> .....	18
<i>Installing</i> .....	20
<i>Assigning Users and Roles</i> .....	21
<i>Azure Resource Names</i> .....	23
Upgrading in Azure .....	23
<i>Azure Backup and Restore</i> .....	23
<i>Locating Azure Ids and Names</i> .....	23
<i>PowerShell</i> .....	24
<i>Updating</i> .....	25
Automation Portal License .....	26
<i>On-Prem</i> .....	26
<i>Azure</i> .....	26
<i>Uploading License</i> .....	26
<i>License Expiry Notification</i> .....	27
Adding a custom hostname/domain to the Automation Portal. ....	27
<i>Azure Portal</i> .....	28
<i>PowerShell</i> .....	30

Active Directory Authorization .....	30
Mobile App .....	31
<i>Icons</i> .....	31
Rotating Always Encrypted Keys .....	32
<i>On-Prem</i> .....	32
<i>Azure</i> .....	32
Overview of Automation Portal .....	33
Search .....	34
Favorites .....	34
Services .....	34
Requests .....	34
Wiki .....	34
Dashboard .....	35
Admin .....	35
Admin .....	36
Settings .....	36
<i>Settings</i> .....	36
<i>Permissions</i> .....	39
<i>License</i> .....	41
<i>Maintenance</i> .....	41
Services .....	42
<i>Creating a Service</i> .....	42
<i>Service Details</i> .....	43
<i>Creating an Offering</i> .....	43
<i>Creating an Offering Folder</i> .....	44
<i>Offering Details</i> .....	45
<i>Moving an Offering</i> .....	46
<i>Offering Markdown</i> .....	47
<i>Adding a Field</i> .....	48
<i>Actions</i> .....	50
Lists .....	51
<i>Creating a List</i> .....	51
<i>List Details</i> .....	51
<i>Creating a List Value</i> .....	52
<i>Actions</i> .....	52
Queries .....	53
<i>Security Recommendations</i> .....	53
<i>Creating a Query</i> .....	54
<i>Parameters and Outputs</i> .....	55
<i>Actions</i> .....	58

Log .....	59
<i>Log Details</i> .....	59
Wiki.....	59
<i>Creating a Folder</i> .....	60
<i>Creating a Doc</i> .....	60
Filters .....	61
Import/Export.....	61
<i>Exporting</i> .....	61
<i>Import</i> .....	62
Group Browser .....	62
Markdown Syntax.....	64
Azure.....	64
<i>Client Secret</i> .....	64
<i>Storage</i> .....	65
Global Search .....	68
Search View .....	69
Services View .....	70
Service Offerings.....	70
New Request .....	70
Request Details.....	71
<i>Stay On Request</i> .....	72
<i>Offering Pre-Fill URL</i> .....	72
Requests View .....	74
Views .....	74
Query Builder.....	74
<i>Building a Query</i> .....	74
<i>Saving a Query</i> .....	75
<i>Using a Saved Query</i> .....	75
Request Details.....	75
Request Fields.....	75
Request Actions.....	76
Multi-Requests Actions .....	76
Field Data.....	77
Dashboard.....	77
Period .....	78
Last Updated.....	79
Currency Symbol.....	79
Chart Color.....	79

Cost and Time Saved .....	80
<b>Customizing .....</b>	<b>82</b>
CSS .....	82
<b>Automation Portal Integration Methodology .....</b>	<b>82</b>
Introduction.....	82
Overview.....	82
Generic Dispatcher 'vs' runbook specific dispatchers.....	83
Requests table .....	83
Example Deployment using Orchestrator. ....	84
A simple request monitoring runbook .....	84
Identifying a new request.....	85
Extracting the request data. ....	86
Updating the Request status .....	90
Best Practice - Single Dispatcher runbook for multiple services and offerings.....	91
<b>Support and Guidance.....</b>	<b>94</b>
<b>Sending Feedback .....</b>	<b>94</b>

## Trademarks

---

App Store is a trademark of Apple Inc., registered in the U.S. and other countries and regions.

IOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

## Using this Guide

---

This user guide is separated into two distinct sections:

- Installing and Administering the Automation Portal
- Automation Portal Integration Methodology

The first section provides guidance on how to install, configure and then use the self-service capabilities of the Automaton Portal.

The second section provides guidance on how to detect and process the Requests raised in the Automation Portal using your automation tool.

## Accessibility Statement

---

The [Web Content Accessibility Guidelines \(WCAG\)](#) defines requirements for designers and developers to improve accessibility for people with disabilities. It defines three levels of conformance: Level A, Level AA, and Level AAA. The Automation Portal is partially conformant with WCAG 2.1 level AA.

# Installing and Administrating the Automation Portal

## System Requirements

The Kelverion Automation Portal has the following system requirements:

### On-Prem

- Windows Server 2019 or Windows Server 2022
- SQL Server 2017, SQL Server 2019, or SQL Server 2022
- Web Server (IIS) Role (using HTTPS)
  - Web Server
    - Security
      - Windows Authentication Role
    - Application Development
      - ASP.NET 4.7 (Windows Server 2019) / ASP.NET 4.8 (Windows Server 2022) Role
- Microsoft .NET Framework 4.8
- Microsoft SQL Server 2012 Native Client – QFE  
(<https://www.microsoft.com/en-us/download/details.aspx?id=50402>)

### Azure

**Note:** Azure expenses will be incurred. The App Service pricing tier is set F1, and the SQL Database pricing tier is set to Basic. Please review your App Service and SQL Database plans and adjust them in Azure to suit your needs.

- Azure subscription
- [PowerShellGet PowerShell Module](#)

**Important:** A Client Secret is created that expires in 12 months. See [Updating Client Secret](#) for further details.

**An Always Encrypted Certificate is created that expires in 12 months.** This is saved to the folder the PowerShell script is running from with the name **Automation Portal.pfx**. Follow best practices to backup and secure this file.

To install the latest version, run the following cmdlet.

```
Install-Module -Name PowerShellGet -Force
```

- [Az PowerShell Module](#) required version 10.3.0

To locate the current version, run the following cmdlet.

```
Get-InstalledModule -Name Az
```

To install the required version, run the following cmdlet.

```
Install-Module -Name Az -Force -RequiredVersion 10.3.0
```



- [SqlServer PowerShell Module](#) required version 22.1.1.

To locate the current version, run the following cmdlet.

```
Get-Module -ListAvailable -Name "SqlServer"
```

To install the required version, run the following cmdlet.

```
Install-Module -Name SQLServer -Force -RequiredVersion 22.1.1
```

- [Microsoft Graph PowerShell Module](#) required version 2.5.0

To locate the current version, run the following cmdlet.

```
Get-Module -ListAvailable -Name "Microsoft.Graph"
```

To locate the required version, run the following cmdlet.

```
Install-Module -Name Microsoft.Graph -Force -RequiredVersion 2.5.0
```

## Web Browser

- Google Chrome 114.0
- Firefox 114.0
- Microsoft Edge 114.0
- JavaScript and Cookies enabled.

## Backup

We recommend you make a backup of your Automation Portal web server and database before upgrading your On-Prem or Azure installation. Please see Microsoft SQL Server documentation for further information.

- [Backup up your app in Azure](#)

## Installing On-Prem

Once you have determined that your server satisfies the installation requirements, you can follow these steps to install the Kelverion Automation Portal on your server.

1. Launch the Kelverion Automation Portal Setup (Kelverion.Web.AutomationPortal.OnPrem.Installer.exe). The **Kelverion Automation Portal Setup Wizard** appears.
2. Review the **End-User License Agreement** and check **I agree to the license terms and conditions**.
3. Click **Install**.
4. Click **Next**.
5. Click **Next** to install Automation Portal to the default folder.

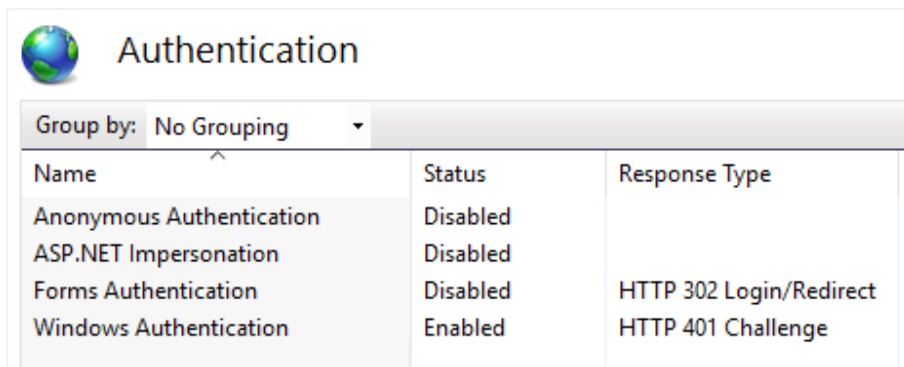
6. Specify the **Server** name or IP address of the SQL Server that will host the portal database. You may enter server\instance.
  7. Specify the **Port**. The default is 1433.
  8. Select **Windows** or **SQL Authentication**. **Note:** this will be used for both database creation and the IIS database connection string.
    - a. If **Windows** is selected, the currently logged on user must have the proper rights on the SQL Server to create a new database.
    - b. If **SQL Authentication** is selected, specify the **Login** and **Password**. This user must have the proper rights on the SQL server to create a new database.
  9. Click **Test Connection**.
  10. See the “Connection Successful” message and click **Ok**.
  11. Click **Next**.
  12. Select **New Database** or **Existing Database** and click **Next**.
    - a. If **New Database** is selected, specify a database name, or leave the default.
    - b. If **Existing Database** is selected, select an existing Automation Portal database from the list.
- Note:** When hosting the database in the Cloud, expenses will be incurred. Please review and confirm that the tier matches your cost and usage requirements.
13. Specify the Web Server **Port**.
  14. Select **Application Pool Identity** or **Custom account** and click **Next**.
    - a. If **Application Pool Identity** is selected, the application pool will be created with the specified identity. The default **applicationPoolIdentity** is recommend.
    - b. If **Custom account** is selected, specify the domain user. The application pool will use this custom identity.
  15. Specify the **AD User Group** and the **AD Admin Group**, then click **Next**. **Note:** only valid when New Database is selected.
    - a. **AD User Group** is the Active Directory group you wish to assign basic user privileges to.
    - b. **AD Admin Group** is the Active Directory group you wish to assign administrative privileges to.
  16. Click **Install**.
  17. Click **Yes** to allow the files to be installed.
  18. Click **Finish** to complete the installer.
  19. Configure the Portal in IIS. See [Internet Information Services \(IIS\) settings](#).
  20. Upload a license to the Portal, see [Uploading License](#).
  21. [Enable Always Encrypted](#) columns.

## Internet Information Services (IIS) settings

### Authentication

The on-prem Automation Portal requires that Windows Authentication be turned on in IIS. After installing the portal, verify these values are correct.

1. Open the **Internet Information Services (IIS) Manager**.
2. Locate and select the **Kelverion Automation Portal** in **Sites**.
3. Locate **Authentication** in IIS settings and double click.
4. Enable **Windows Authentication**.
5. Disable **Anonymous**, **ASP.Net** and **Forms Authentication**.



Name	Status	Response Type
Anonymous Authentication	Disabled	
ASP.NET Impersonation	Disabled	
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Enabled	HTTP 401 Challenge

### HTTPS in IIS

HTTPS must be configured for the Automation Portal to function.

To enable HTTPS in IIS please refer to [How to Set up an HTTPS Service in IIS](#).

### Enable Always Encrypted

By default, sensitive data columns in the Automation Portal database are not encrypted and require additional configuration. **Always Encrypted** needs to be configured for the **[dbo].[Query].[Password]** and **[dbo].[SecureRequestData].[Value]** columns. If **[dbo].[Query].[Password]** has already been encrypted in a previous version, you only need to configure encryption for **[dbo].[SecureRequestData].[Value]**.

1. On the Automation Portal IIS Server, open **SQL Server Management Studio** as **Run as Administrator**. If your SQL Server Management Studio is not located on the IIS Server, you will later have to import to the IIS Server, the certificates that were generated on the SQL Server Management Studio machine.
2. Using **SQL Server Management Studio** connect to the SQL Server that is hosting the portal database.
3. Right click the Automation Portal database and click **Tasks**.
4. Select **Encrypt Columns....**
5. The Always Encrypted dialog opens. Click Next to move to Column Selection

6. Configure **dbo.Query** table encryption. This may not be necessary if upgrading from a previous version where this column has already been encrypted.
    - a. Select **Password** column in the **dbo.Query** table.
    - b. Select Encryption Type: **Deterministic**.
    - c. Set Encryption Key: **CEK\_Auto1 (New)**.
  7. Configure **dbo.SecureRequestData** table encryption.
    - a. Select **Value** column in the **dbo.SecureRequestData** table.
    - b. Select Encryption Type: **Deterministic**.
    - c. Set Encryption Key: **CEK\_Auto1 (New)**.
  8. Click **Next**.
  9. Select column master key: **Auto generate column mastery**. (Not applicable if encryption is already configured).
  10. Select the key store provider: **Windows Certificate store** (Not applicable if encryption is already configured).
  11. Select a master key source depending on your **IIS Application Pool Identity** (Not applicable if encryption is already configured). You can view your configured IIS Application Pool Identity under **Application Pools > AutomationPortal > Advanced Settings > Process Model > Identity**.
    - a. Select **Current User** if your **IIS Application Pool Identity** is.
      - **Custom identity**
    - b. Select **Local Machine** if your **IIS Application Pool Identity** is.
      - **ApplicationPoolIdentity**
      - **LocalSystem**
      - **NetworkService**
- Note:** If “Local Machine” is not a selectable option move to step 1 and make sure to launch **SQL Server Management Studio** as an Administrator.
12. Click **Next**.
  13. Select **Proceed to finish now**.
  14. Click **Next**.
  15. Click **Finish**.
  16. Click **Close**.
  17. Import the newly generated **Always Encrypted Auto Certificate** on the **IIS server**. This step is only applicable if **SQL Server Management Studio** steps 1 to 15 were not performed on the IIS Server machine and if the encryption certificate was not previously imported. For details, please refer to [Always Encrypted Certificate Import/Export](#).

- a. Export the **Always Encrypted Auto Certificate** created by **SQL Server Management Studio**.
  - b. Import the certificate on the **IIS Server**.
18. Open the **Web.config** located on the **IIS server** in the Automation Portal installation folder in a text editor.
19. Append “;**Column Encryption Setting=Enabled**” to the end of <connectionString>
20. Configure IIS to use the imported database encryption certificate imported in step 17. This depends on your **IIS Application Pool Identity**. You can view your configured IIS Application Pool Identity under **Application Pools > AutomationPortal > Right click - Advanced Settings > Process Model > Identity**.
  - a. If your **IIS Application Pool Identity** is **Custom** identity:
    - i. In Advanced Settings under **Process Model** click the ellipsis next to **Identity**. Select **Custom account** and specify the user account used to store the database encryption certificate, in the form <domain>\<user>.
  - b. If your **IIS Application Pool Identity** is **ApplicationPoolIdentity**, **LocalSystem**, or **NetworkService**
    - i. Open the **Always Encrypted Auto Certificate** that was imported on the IIS Server. This should be located under **Certificates (Local Computer) > Personal > Certificates**.
    - ii. Right click the **Always Encrypted Auto Certificate > All Tasks > Manage Private Keys**.
    - iii. Click **Add...** and change **Locations** from domain to the local computer.
    - iv. Add an account. Select one of the following.
      - Add the **IIS AppPool\AutomationPortal** account if the **IIS Application Pool Identity** is **ApplicationPoolIdentity**,
      - Add the **LocalSystem** account if the **IIS Application Pool Identity** is **LocalSystem**, or
      - Add the **NETWORK SERVICE** account if the **IIS Application Pool Identity** is **NetworkService**.
21. Restart the Automation Portal in IIS.

## Always Encrypted Certificate Import/Export

To export an **Always Encrypted** certificate created by **SQL Server Management Studio**:

1. On the **SQL Server Management Studio** machine, select **Run** from the Start menu.
2. Enter **mmc**.
3. Click **OK**.
4. Open **File** menu.
5. Select **Add/Remove Snap-in...**
6. Select **Certificates**.
7. Click **Add**.

8. Select certificates account depending on your **IIS Application Pool Identity**.
  - a. Select **My user account** if your IIS Application Pool Identity is.
    - **Custom**
  - b. Select **Computer account – Local Computer** if your IIS Application Pool Identity is.
    - **ApplicationPoolIdentity**
    - **LocalSystem**
    - **NetworkService**
9. Click **Finish**.
10. Click **OK**.
11. Expand **Personal** folder.
12. Select **Certificates** folder.
13. Right click the newly created **Always Encrypted Auto Certificate**
14. Select **All Tasks**.
15. Select **Export...**
16. Click **Next**.
17. Select **Yes, export the private key**.
18. Click **Next**.
19. Select **Personal Information Exchange – PKCS #12(.PFX)**.
  - a. Select **Include all certificates in the certification path if possible**.
  - b. Select **Export all extended properties**.
  - c. Select **Enable certificate privacy**.
20. Click **Next**.
21. Select **Password**.
22. Enter a **Password**.
23. Click **Next**.
24. Enter a **File name**.
25. Click **Next**.
26. Click **Finish**

To import a certificate on the **IIS Server**:

1. On the **IIS Server**, select **Run** from the Start menu.

2. Enter **mmc**.
3. Click **OK**.
4. Open **File** menu.
5. Select **Add/Remove Snap-in...**
6. Select **Certificates**.
7. Click **Add**.
8. Select certificates account depending on your **IIS Application Pool Identity**.
  - a. Select **My user account** if your IIS Application Pool Identity is.
    - **Custom**
  - b. Select **Computer account – Local Computer** if your IIS Application Pool Identity is.
    - **ApplicationPoolIdentity**
    - **LocalSystem**
    - **NetworkService**
9. Click **Finish**.
10. Click **OK**.
11. Right click the **Personal** folder.
12. Select **All Tasks**.
13. Select **Import...**
14. Click **Next**.
15. Click **Browse** for the exported certificate.
16. Select exported .pfx file.
17. Click **Open**.
18. Click **Next**.
19. Enter certificate **Password**.
20. In the Import options section, check the box **Mark this key as exportable**.
21. Click **Next**.
22. Select **Automatically select the certificate store based on the type of certificate**.
23. Click **Next**.
24. Click **Finish**.

## High Availability (HA) Deployment

High availability (HA) has been evaluated and is officially supported specifically when using the Windows Server Network Load Balancing (NLB) feature as detailed below.

**Note:** Additional licensing fees apply. Please speak to your account manager regarding support for other HA configurations.

Additional System Requirements:

- **Network Load Balancing** feature installed.
  - Each Server in the cluster must have a matching version of ASP.NET.
1. Configure a cluster of IIS servers by installing the NLB Windows Server Feature. <https://docs.microsoft.com/en-us/windows-server/networking/technologies/network-load-balancing>
  2. Complete the On-Prem install steps above on the first Host (HostA) in the Cluster.
  3. In each additional host:
    - a. Copy the folder **C:\Inetpub\Kelverion Automation Portal** from HostA to all other hosts in the cluster, in the same location **C:\Inetpub\Kelverion Automation Portal**.
    - b. Open IIS
    - c. Right click on Sites and select **Add Website**.
    - d. Enter Site name **AutomationPortal**.
    - e. Select Physical path **C:\Inetpub\Kelverion Automation Portal**.
    - f. Adjust any other settings to match HostA.
    - g. Click **OK**.
  4. On HostA the installer may automatically grant SQL permissions to the Portal database. You may need to manually add permissions to the Portal database for the additional hosts.
  5. Complete the install tasks in the [Internet Information Services \(IIS\) settings](#) section on each host.
  6. From a supported web browser, navigate to the Portal targeting each host to confirm a working setup.
  7. Upload a license to the Portal, see [Uploading License](#).

## Additional Configuration

The following are user configurable requirements that are not required to install or run the Automation Portal but are recommend for securing your Automation Portal.

### *IIS Disable Server Header*

Follow these steps to disable server header for IIS.

1. Open regedit.
2. Navigate to Computer\HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\HTTP\Parameters



3. If DisableServerHeader does not exist, create it (DWORD 32bit), and give it a value of **2**. If it does exist, and the value is not 2, set it to 2.
4. Reboot the server.

### SSL/TLS Encryption

Weak Ciphers and Protocols should be disabled. Please review the Microsoft [TLS/SSL overview](#) page and disable what is appropriate for your organization.

## Upgrading On-Prem

To upgrade the previous version of the Automation Portal, you can follow these steps to upgrade the Kelverion Automation Portal on your server.

**Note:** The Kelverion Automation Portal 3.6 or higher must be installed to perform an upgrade to version 3.8.

1. Launch the Kelverion Automation Portal Setup (Kelverion.Web.AutomationPortal.OnPrem.Installer.exe). The **Kelverion Automation Portal Setup Wizard** appears.
2. Review the **End-User License Agreement** and check **I agree to the license terms and conditions**.
3. Click **Install**.
4. Click **Next**.
5. Click **Next** to install Automation Portal to the default folder.
6. Specify the **Server** name or IP address of the SQL Server that will host the portal database. You may enter server\instance.
7. Specify the **Port**. The default is 1433.
8. Select **Windows** or **SQL Authentication**. **Note:** this will be used for both database creation and the IIS database connection string.
  - a. If **Windows** is selected, the currently logged on user must have the proper rights on the SQL Server to create a new database.
  - b. If **SQL Authentication** is selected, specify the **Login** and **Password**. This user must have the proper rights on the SQL server to create a new database.
9. Click **Test Connection**.
10. See the **Connection Successful** message and click **Ok**.
11. Click **Next**.
12. Select **Existing Database** and click **Next**.
13. Specify the Web Server **Port**.
14. Select **Application Pool Identity** or **Custom account** and click **Next**.
  - a. If **Application Pool Identity** is selected the application pool will be created with the specified identity. The default **applicationPoolIdentity** is recommend.

- b. If **Custom account** is selected specify the domain user. The application pool will use this custom identity.
15. Click **Install**.
16. Click **Yes** to allow the files to be installed.
17. Click **Finish** to complete the installer.
18. Configure the Portal in IIS. See [Internet Information Services \(IIS\) settings](#).
19. [Enable Always Encrypted](#) columns for **[dbo].[Query].[Password]** and **[dbo].[SecureRequestData].[Value]** columns. Verify that both columns have been configured for Always Encrypted. If **[dbo].[Query].[Password]** has previously been encrypted, follow the steps to encrypt **[dbo].[SecureRequestData].[Value]**.

## Installing in Azure

The Kelverion Automation Portal can be installed in Azure using PowerShell. This script will install the Automation Portal App Service and create and configure a SQL database in the Azure Subscription you specify.

**Notes:** Azure expenses will be incurred. The App Service pricing tier is set B1, and the SQL Database pricing tier is set to Basic. Please review your App Service and SQL Database plans and adjust them in Azure to suit your needs.

## Locating Azure Ids

You will need to have your Azure Tenant Id and Subscription Id. These ids can be in the following locations in the Azure portal.

- **Tenant Id** - The tenant id can be in **Azure Active Directory** -> **Properties** -> **Directory ID**.
- **Subscription Id** - The subscription id can be in **Subscriptions** -> **<Your subscription>** -> **Overview** -> **Subscription ID**.

## PowerShell

The **Start-AutomationPortalInstall.ps1** script can be used to deploy the Automation Portal to Azure. The script will create a new Resource Group in Azure with a new SQL Server, SQL Database, Web App and register the Web App with Azure Active Directory.

### *Start-AutomationPortalInstall.ps1*

#### Syntax

```
Start-AutomationPortalInstall -TenantId <String> -SubscriptionId <String> -ResourceGroupName
<String> -Location <String> -SqlAdminCredentials <PSCredential> [-ClientSecretExpiresInMonths
<int>] [-Environment <String>]
```

#### Description

Installs the Kelverion Automation Portal in Azure.

#### Parameters

- TenantId <String>

Specifies the Azure tenant id. The application resources will be created inside this tenant.

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters	false

- [SubscriptionId <String>](#)

Specifies the Azure subscription id.

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters	false

- [ResourceGroupName <String>](#)

Specifies the Azure resource group name that will be created.

Required?	true
Position?	named
Default Value	KelverionAutomationPortal
Accept Pipeline Input?	false
Accept Wildcard Characters	false

- [Location <String>](#)

Specifies the Azure Resource Location where the portal will be deployed. Example: East US

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters	false

- [SqlAdminCredential <PSCredential>](#)

Specifies the username and password that will be set for the created SQL Database.

Required?	true
Position?	named

Default Value	None
Accept Pipeline Input?	false
Accept Wildcard Characters	False

#### - ClientSecretExpiresInMonths <int>

Specifies the length in months when the client secret will expire.

Required?	false
Position?	named
Default Value	12
Accept Pipeline Input?	false
Accept Wildcard Characters	false

#### - Environment <String>

Specifies the Azure environment. Use the default value unless you are targeting an Azure government or similar.

Required?	false
Position?	named
Default Value	AzureCloud
Accept Pipeline Input?	false
Accept Wildcard Characters	false

## Installing

The following steps can be used to install and configure the Portal in Azure.

**Important:** A [Client Secret](#) is created that expires in 12 months.

An **Always Encrypted Certificate** is created that expires in 12 months. This is saved to the folder the PowerShell script is running from with the name **Automation Portal.pfx**. Follow best practices to backup and secure this file.

1. Extract the **Kelverion.Web.AutomationPortal.Azure.Installer.zip**
2. Verify that you have installed the version of the PowerShell modules listed in System Requirements.
3. Launch a fresh instance of **Windows PowerShell ISE** with administrator rights.
4. Change directory to the extracted **Kelverion.Web.AutomationPortal.Azure.Installer** folder.
5. Run **Start-AutomationPortalInstall.ps1** with the appropriate parameters for your Azure environment. During installation you will be prompted for credentials two times. First, during **Step 1: Deploying App Service** and then during **Step 2: Installing App Registration**. Use the appropriate credentials for each step. Follow the example below that is appropriate for your environment.

```
# Template from running the Start-AutomationPortalInstall.ps1 script
$TenantId = "PUT YOUR TENANT ID HERE"
$SubscriptionId = "PUT YOUR SUBSCRIPTION ID HERE"
$ResourceGroupName = "PUT YOUR RESOURCE GROUP NAME HERE"
$Location = "PUT YOUR LOCATION NAME HERE" #Examples: East US, Canada Central
$SqlAdminCredential = Get-Credential -Message "New SQL Admin Credentials"

.\Start-AutomationPortalInstall.ps1 `
    -TenantId $TenantId `
    -SubscriptionId $SubscriptionId `
    -ResourceGroupName $ResourceGroupName `
    -SqlAdminCredential $SqlAdminCredential `
    -Location $Location
```

- Wait for the PowerShell script to report **Installation Complete**.

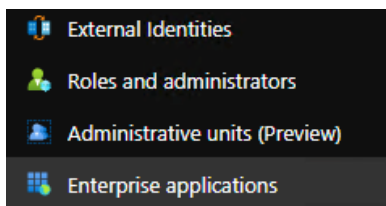
**Important:** A **Client Secret** is created that expires in 12 months.

An **Always Encrypted Certificate** is created that expires in 12 months. This is saved to the folder the PowerShell script is running from with the name **Automation Portal.pfx**. Follow best practices to backup and secure this file.

- Copy the URL for **App Host Name** and paste in a browser to open the portal. Login using the same user that ran the install.
- Install a valid [license](#).
- Assign access to additional users.

## Assigning Users and Roles

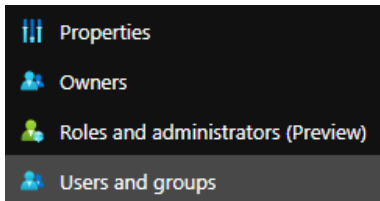
- In the Azure Portal, select **Azure Active Directory**.
- Select **Enterprise applications**.



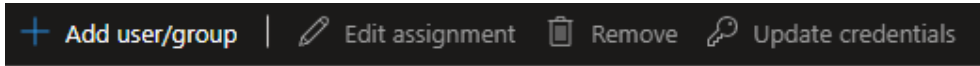
- Select **All applications** and select the Automation Portal application. You also can search 'kelverion' and get the list of Kelverion applications.



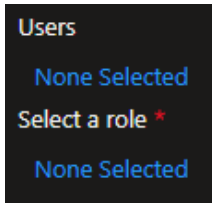
4. Select **Users and groups**.



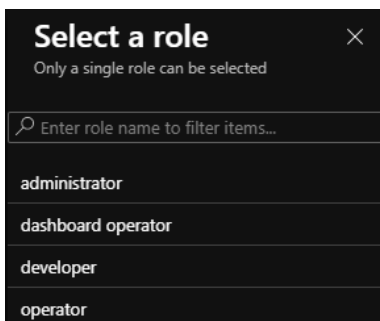
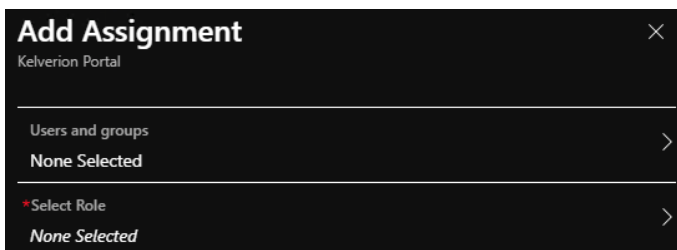
5. Click **Add user**.



6. Select **Users**, assign user(s) to the application, and click **Select**. You can also select groups of users rather than individual users. Note: Nested Groups are not supported for roles in Azure currently



7. Select **Role** and assign a role to the selected user(s) or to a group with multiple users. The portal application offers the following roles and click **Select**.
- **Administrator** - Administrator can create Services and customize the portal.
  - **Dashboard operator** – Dashboard users have basic user permissions plus permission for the Dashboard.
  - **Developer** – Developer users have access to build Services, Lists and Queries.
  - **Operator** - Basic operator and has no privileges to create new Services or customize the portal.



8. Click **Assign**.

## Azure Resource Names

The Automation Portal for Azure installs the following resources in Azure.

- App Service
- Azure Active Directory App Registration
- SQL Server
- SQL Database
- App Service Plan

To support future updates the names of these resources should not be changed. Consider using [Azure Tags](#) to organize your Azure resources.

## Upgrading in Azure

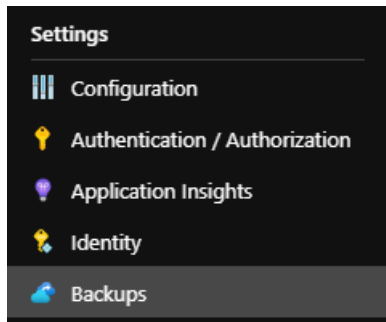
The Kolverion Automation Portal can be upgraded in Azure using PowerShell.

**Note:** The Kolverion Automation Portal 3.6 or higher must be installed to perform an upgrade to version 3.8.

**Note:** If your Portal App Service is configured using F1 or D1 tier it will be upgraded to B1 tier.

## Azure Backup and Restore

Before upgrading the Automation Portal in Azure, it is best practice to create an Azure App Service backup. A backup of the Automation Portal Azure App Service and the SQL Database can be created in the **Settings** -> **Backup** page of the App Service.



For more information about configuring, creating, or restoring a backup please review [Back up your app in Azure](#) available from Microsoft.

## Locating Azure Ids and Names

You will need to have your Azure Tenant Id, Subscription Id, Resource Group Name and Web App Name. These ids and names can be in the following locations in the Azure portal.

- **Tenant Id** – The tenant id can be in **Azure Active Directory** -> **Properties** -> **Directory ID**.
- **Subscription Id** – The subscription id can be in **Subscriptions** -> <Your subscription> -> **Overview** -> **Subscription ID**.
- **Resource Group Name** – The resource group that contains your web app can be in **Resource groups**.

- **App Service Name** – The web app name can be in the Resource group. **Resource groups** -> <Your resource group>. The Automation Portal App Service will have the type of **App Service**.

## PowerShell

The provide **Update-AutomationPortalAppService.ps1** script can be used to update the Automation Portal in Azure. The script will update the web app in the specified Resource Group in Azure and update the database in SQL Server.

### Update-AutomationPortalAppService.ps1

#### Syntax

```
Update-AutomationPortalAppService -TenantId <String> -SubscriptionId <String> -AppServiceName <String> [$Environment <String>]
```

#### Description

Updates the Kelverion Automation Portal in Azure.

#### Parameters

##### - TenantId <String>

Specifies the Azure tenant id.

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters	false

##### - SubscriptionId <String>

Specifies the Azure subscription id.

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters	false

##### - AppServiceName <String>

Specifies the Azure App Service Name.

Required?	true
Position?	named
Default Value	none



Accept Pipeline Input?	false
Accept Wildcard Characters	false

#### - Environment <String>

Specifies the Azure environment.

Required?	false
Position?	named
Default Value	AzureCloud
Accept Pipeline Input?	false
Accept Wildcard Characters	false

## Updating

The following steps can be used to update the Portal in Azure. During upgrading, the Web App will be stopped during updating and will be started at the end of the update.

**Note:** The Kelverion Automation Portal 3.6 or higher must be installed to perform an update to version 3.8.

1. Extract the **Kelverion.Web.AutomationPortal.Azure.Updater.zip** file.
2. Verify that you have installed the version of the PowerShell modules listed in System Requirements.
3. Launch a fresh instance of **Windows PowerShell ISE** with administrator rights.
4. Change directory to extracted **Kelverion.Web.AutomationPortal.Azure.Updater** folder.
5. Run **Update-AutomationPortalAppService.ps1** with the appropriate parameters for your Azure environment. The script will prompt you for your Azure credentials. Follow the example below.

```
# Template from running the Update-AutomationPortalAppService.ps1 script
$TenantId = "PUT YOUR TENANT ID HERE"
$SubscriptionId = "PUT YOUR SUBSCRIPTION ID HERE"
$AppServiceName = "PUT YOUR APP SERVICE NAME HERE"
.\Update-AutomationPortalAppService.ps1 `
  -TenantId $TenantId `
  -SubscriptionId $SubscriptionId `
  -AppServiceName $AppServiceName
```

6. Wait for the PowerShell script to complete.
7. By default, API permission and user credential are applied to the upgraded version as per the previous version of the portal.

8. Install the valid license and you are good to use Portal service on Azure.

## Automation Portal License

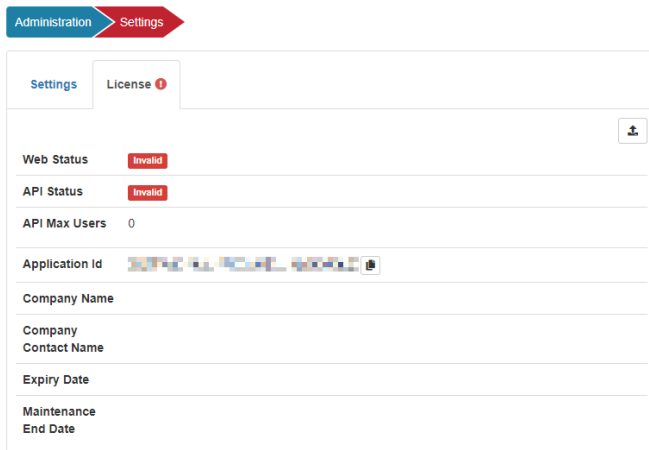
After installing the portal, you must provide a valid Kelverion license to activate the Portal and REST API (Azure only).

### On-Prem

For an on-prem install, provide the fully qualified domain name (FQDN) of your IIS Server to Kelverion when requesting a license.

### Azure

For an Azure install, provide the Application ID when requesting a license. You can find the Application Id value in **Administration -> Settings -> License** tab of the portal.



Administration Settings

Settings License

Web Status Invalid

API Status Invalid

API Max Users 0

Application Id

Company Name

Company Contact Name

Expiry Date

Maintenance End Date

## Uploading License

1. Click the **Admin** gear.
2. Click **Settings**.
3. Click the **License** tab.
4. Click the **Upload** in the **License** section to upload a license.
5. Select a KAL file.
6. Click **Upload**.

Administration > Settings

Settings License ⓘ

Web Status Invalid

API Status Invalid

API Max Users 0

Application Id [Long alphanumeric string]

Company Name

Company Contact Name

Expiry Date

Maintenance End Date

- Once you upload the valid license, review any licensing alerts before continuing.

**Licensing:** If applicable please update your Kelverion Integration Pack and Integration Module licenses.

## License Expiry Notification

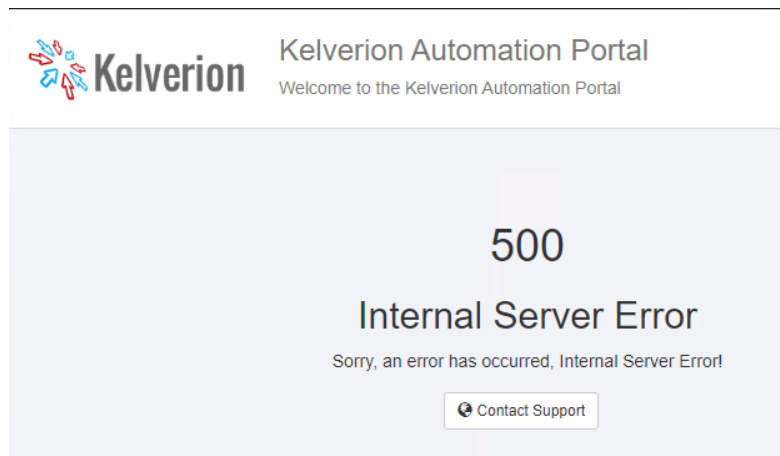
Administrators will see a notification displayed when the license is due to expire in 30 days or less.

**Licensing:** Your portal license will expire in 29 days. Please contact [Kelverion support](#) for assistance.

## Adding a custom hostname/domain to the Automation Portal.

Adding a custom hostname to the automation portal is a 2-stage process. In the first stage, you must configure DNS and bind the new hostname to the Azure App Service. The following tutorial from Microsoft will guide you through those steps in [detail](#).

Once the hostname has been correctly configured, you should be able to connect to the new address. However, as the configuration is not complete, you will see that you are redirected to the old address, and you see the Portal error page.



To resolve this, you must perform the second stage of the configuration, updating the Azure AD App registration so that the users can correctly authenticate.

**Note:** The app configuration format is extremely specific, a misplaced (or an additional) in the wrong place will cause the portal pages to fail with the error page above.

The example below assumes that we are using the hostname *portal.kelverion.com*.

There are four properties that must be configured on the Azure AD Application.

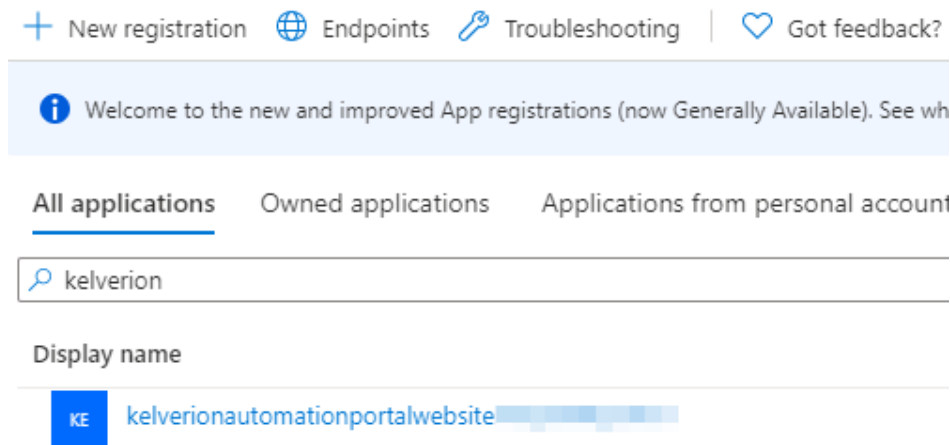
#### Custom Domain Values

Property	Example Value
1 IdentifierUri / App ID URI	<a href="https://portal.kelverion.com/">https://portal.kelverion.com/</a>
2 HomePage	<a href="https://portal.kelverion.com/">https://portal.kelverion.com/</a>
3 LogoutUrl	<a href="https://portal.kelverion.com/Account/EndSession">https://portal.kelverion.com/Account/EndSession</a>
4 ReplyUrls	<a href="https://portal.kelverion.com/">https://portal.kelverion.com/</a>

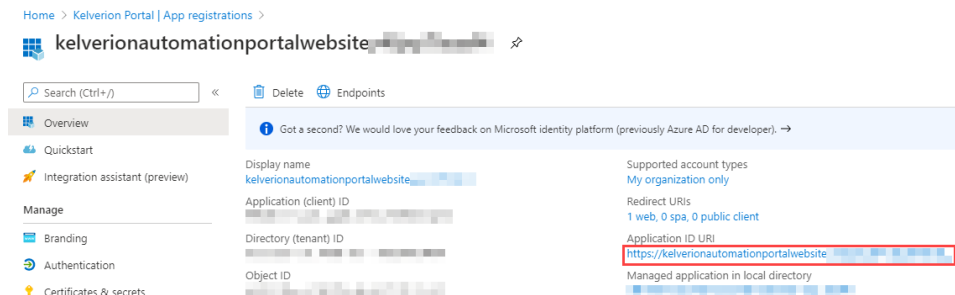
## Azure Portal

In the Azure Portal navigate to the **Azure AD Blade** and select **App registrations**.

1. In the search box type **Kelverion**.



2. Click on the portal name.



3. In the **Expose an API** dialog, update the **Application ID URI** with **IdentifierUri** from the **Custom Domains Table** above.

Search (Cmd+/) <<

Overview  
Quickstart  
Integration assistant (preview)

**Manage**

Branding  
Authentication  
Certificates & secrets  
Token configuration  
API permissions  
**Expose an API**  
Owners  
Roles and administrators (Preview)

Application ID URI

Scopes defined by this API

Define custom scopes to restrict access to data and functionality protect API can request that a user or admin consent to one or more of these.

[+ Add a scope](#)

Scopes	Who can
<a href="https://kelverionautomationportalwebsitev2ev2ysueh...">https://kelverionautomationportalwebsitev2ev2ysueh...</a>	Admins

Authorized client applications

Authorizing a client application indicates that this API trusts the application this API.

4. In the **Branding** dialog, update **Home page URL** with **HomePage** from the **Custom Domains Table** above.

Overview  
Quickstart  
Integration assistant (preview)

**Manage**

**Branding**  
Authentication  
Certificates & secrets  
Token configuration  
API permissions  
Expose an API

Name \*


Logo

Upload new logo

Home page URL

Terms of service URL

Privacy statement URL

Publisher domain     
The application's consent screen will show 'Unverified publisher domain' [Learn more about publisher domain](#)

5. In the **Authentication** dialog, update **Redirect URIS** with **ReplyURLs** from the **Custom Domains Table** above.

Branding  
**Authentication**  
Certificates & secrets  
Token configuration  
API permissions  
Expose an API  
Owners  
Roles and administrators (Preview)  
Manifest

[+ Add a platform](#)

Web

Redirect URIs


The URIs we will accept as destinations when returning [more about Redirect URIs and their restrictions](#)

[Add URI](#)


6. In the **Authentication** dialog, update **Logout URL** with **LogoutURL** from the **Custom Domains Table** above.


 Overview


 Quickstart

 Integration assistant (preview)


### Manage

 Branding

 Authentication

 Certificates & secrets

 Token configuration

 API permissions

[Add URI](#)

### Logout URL

This is where we send a request to have the application work correctly.

### Implicit grant

Allows an application to request a token directly from

## PowerShell

If you have the Azure AD powershell module installed, the following is an example of making these changes using powershell.

```
$DisplayName = 'Azure AD Application Name' # e.g.
kelverionautomationportalwebsiteabcdefghijkl
$NewName = 'https://host.domain.tld' # e.g. https://portal.kelverion.com

if ($NewName -match "/$"){
    $HomePage = $NewName
}
else{
    $HomePage = "{0}/" -f $NewName
}
$replyUrls = @($HomePage)
$IdentifierUri = @($HomePage)
$LogoutUrl = "{0}Account/EndSession" -f $homepage

Connect-AzureAD
$App = Get-AzureADApplication -Filter $("DisplayName eq '{0}'" -f $DisplayName )

$config = @{
    IdentifierUri = $IdentifierUri;
    LogoutUrl    = $LogoutUrl;
    HomePage     = $HomePage;
    replyUrls    = $replyUrls;
}
$config
Set-AzureADApplication -ObjectId $App.ObjectId @Config -verbose
```

## Active Directory Authorization

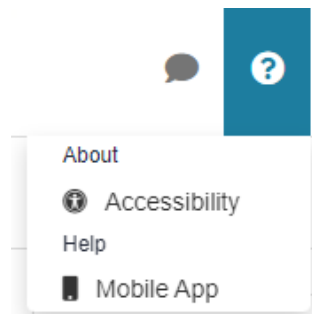
Active directory groups specified for assigning user, administrator and offering permissions must be an active directory security group. Other group types such as distribution groups are not supported.

## Mobile App

The Automation Portal Mobile App is available in the App Store. An API license is required to enable support for the Mobile App.

The Automation Portal Mobile App can be used with the Portal Settings QR Code to quickly configure the mobile app with the Azure Tenant ID, Azure Application ID, and Automation Portal URL.

The Portal Settings QR Code can be in the in the Help Menu. Click on **Mobile App** to display the Portal Settings QR Code.



### Mobile App

#### Automation Portal Mobile App Configuration

Use the **Portal Settings** QR Code to quickly configure the Automation Portal Mobile App.

1. Install the Automation Portal Mobile App from the App Store.
2. Launch the Automation Portal App.
3. Tap **QR Code**.
4. Tap **OK** to allow access to the camera.
5. Scan the **Portal Settings** QR Code to configure the App.

When the configuration is complete the **Sign In** button will appear.



## Icons

The Automation Portal Mobile app requires all Services and Offerings to have upgraded icons. If you have Services and Offerings created using Automation Portal version 3.2 or earlier, you will need to upgrade your icons.

When editing your Services or Offerings you will be presented with the option to upgrade.

**New Icon Set Available.** This offering is using the older icon set. Updating to the new icon set will reset your selected icon to the default icon ⚙️.

Upgrade

## Rotating Always Encrypted Keys

### On-Prem

For details and steps on rotating your current column master key review the [Rotate Always Encrypted keys using SQL Server Management Studio](#) documentation.

### Azure

The **Update-AutomationPortalColumnMasterKey.ps1** script can be used to rotate the current column master key to a new column master key.

#### *Update-AutomationPortalColumnMasterKey.ps1*

##### Syntax

```
Update-AutomationPortalColumnMasterKey -TenantId <String> -SubscriptionId <String> -AppServiceName <String> -CurrentColumnMasterKeyName <String> -NewColumnMasterKeyName <String> -CertificateName <String> [-Environment <String>]
```

##### Description

Rotates the current column master key to a new column master key.

##### Parameters

###### - TenantId <String>

Specifies the Azure tenant id. The application resources will be created inside this tenant.

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters	false

###### - SubscriptionId <String>

Specifies the Azure subscription id.

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters	false

###### - AppServiceName <String>

Specifies the portal app service to update the column master key.

Required?	true
-----------	------



Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters	false

- `CurrentColumnMasterKey <String>`

Specifies the name of the current column master key.

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters	false

- `NewColumnMasterKey <String>`

Specifies the name of the new column master key.

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters	false

- `CertificateName <String>`

Specifies a unique name for the new certificate.

Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters	false

## Overview of Automation Portal

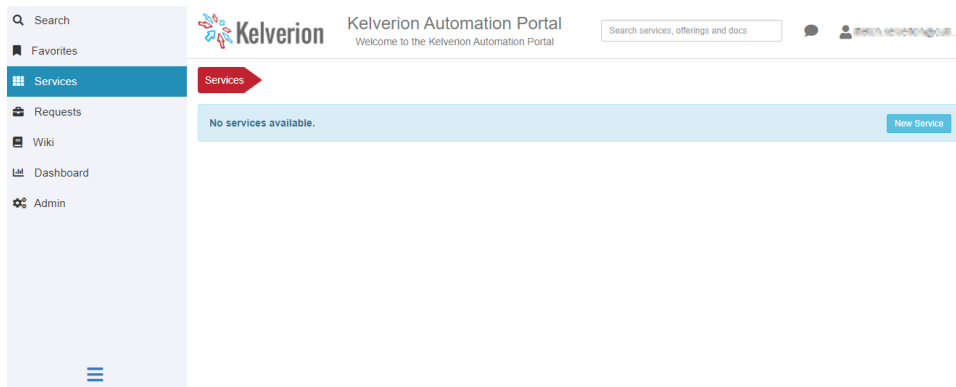
The Kelverion Automation Portal is an easy to implement self-service front end for your automation solutions. It offers a versatile interface without the lengthy list of prerequisites required by many other self-service portals available today.

The Automation Portal ethos is to **keep things simple and flexible**. The portal has been designed to complement the existing Automation platforms and does not attempt to replicate the automation platforms capability. Integration with the automation tools takes place via the Automation portal database and is most easily achieved using the Kelverion integrations for SQL.

The Automation Portal has been designed for use with System Center Orchestrator and Azure Automation in mind, however it equally complements any automation platform or scripting language that can read and write to the Automation Portal database.

The Automation Portal is simple to implement, but even more importantly it is easy for your end users to navigate, and by adding a simple web interface to your solutions you can allow users to easily interact with your automated solutions.

The Automation Portal is organized into seven principal areas, the **Search** view, **Favorites** view, **Services** view, **Requests** view, **Wiki** view, **Dashboard** view and **Admin** view.



## Search

The Search view is where you can search Services, Offerings and Wiki docs.

## Favorites

The Favorites view is where you will see all your custom **Favorite Offerings**.

## Services

The Services view is where you will see all your custom **Services** that contain your defined **Offerings**.

## Requests

The **Requests** view is where you can view your request details and administer your submitted requests.

## Wiki

The **Wiki** view is where you can view Folders and Documents and is an optional view that can be activated in the **Settings** page. Use this view to display valuable information to users.

## Dashboard

The **Dashboard** view is where dashboard users or administrators can view a collection of charts that display important stats.

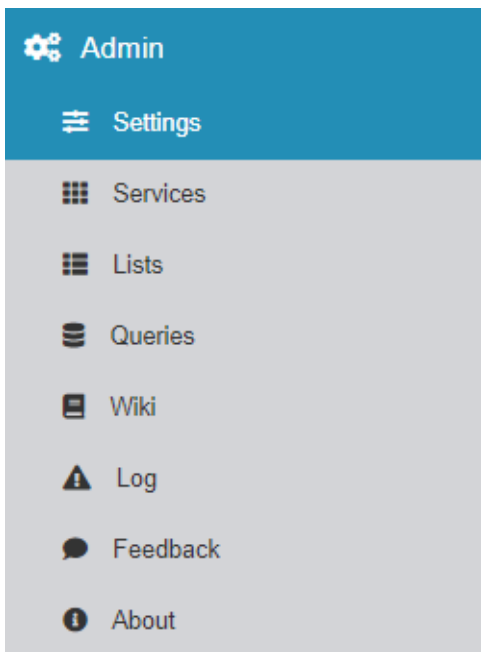
## Admin

The **Admin** view is where the portal administrators create services, offerings and define custom fields for each offering. The Admin menu is only viewable by members in the **AD Admin Group** specified during installation.

## Admin

The Admin menu provides the following:

- **Settings**
  - **Settings** – Used to customize and configure portal settings.
  - **Permissions** – On-Prem only. Used to configure each user group.
  - **License** – Used to license the portal.
  - **Maintenance** – Used to perform maintenance tasks.
- **Services** – Used to define and administer services, offering and fields.
- **Lists** – Used to define reusable lists for radio button or lists.
- **Queries** – Used to define reusable database-based fields.
- **Wiki** – Used to define Wiki folders and documents.
- **Log** – Used to view error log.
- **Feedback** – Used to view user feedback.
- **About** – Information about the Automation portal.



## Settings

The Settings view allows you to customize your portal. Clicking **Edit** will allow you to customize the following:



### Settings

- **Portal Name** – The name of the portal displayed on the main page.
- **Portal Description** – The description of the portal displayed on the main page.
- **Portal Logo** – The brand logo that is displayed on each page.
  - Recommended size of the logo is two hundred pixels wide by fifty pixels high.

- **Support Link** – An option to configure your support link which allows the operator to be re-directed to the configured URL.
- **Enable Wiki** – An option to turn on the Wiki feature. When turned on the **Wiki Name**, menu item will appear at the top of the page next to Requests.
- **Wiki Name** – The name of the Wiki menu item that appears with **Enable Wiki** is turned on.
- **Query Timeout** – The number of seconds used for timeout by the database connection for query-based fields.
- **Currency Symbol** – The currency symbol used on the Dashboard.
- **Stay on Request** – An option to remain on the request page after submitting a request.
- **Enable Themes** – When turned on, the theme will be able to be selected.
- **Attachment Upload Folder**– On-Prem only. Defines the folder path location used by the **File Attachment** field type.
- **Azure Storage Account Connection String** – Azure only (visible only when settings are being updated). Defines the connection information used to access the Azure Storage Container used by the **File Attachment** field type. See [Azure Storage](#) for details of finding the Azure Storage Container name and connection string.
- **Azure Storage Container Name** – Azure only. Defines the Azure Storage Container used by the **File Attachment** field type. See [Azure Storage](#) for details of finding the Azure Storage Container name and connection string.
- **Enable Reason for Rejecting Request** – An option to configure rejecting a request with a reason for rejecting.
- **Dashboard Update Interval (min)** – The refresh interval of the dashboard.


## On-Prem Settings

Administration Settings

Settings	Permissions	License	Maintenance
Portal Name	Kelverion Automation Portal		
Portal Description	Welcome to the Kelverion Automation Portal		
Portal Logo			
Support Link	<a href="https://www.kelverion.com/support">https://www.kelverion.com/support</a>		
Enable Wiki	True		
Wiki Name	Wiki		
Query Timeout (sec)	10		
Currency Symbol	£		
Stay On Request	False		
Enable Themes	True		
Attachment Upload Folder			
Enable Reason For Rejecting Request	True		
Dashboard Update Interval (min)	2		

## Azure Settings

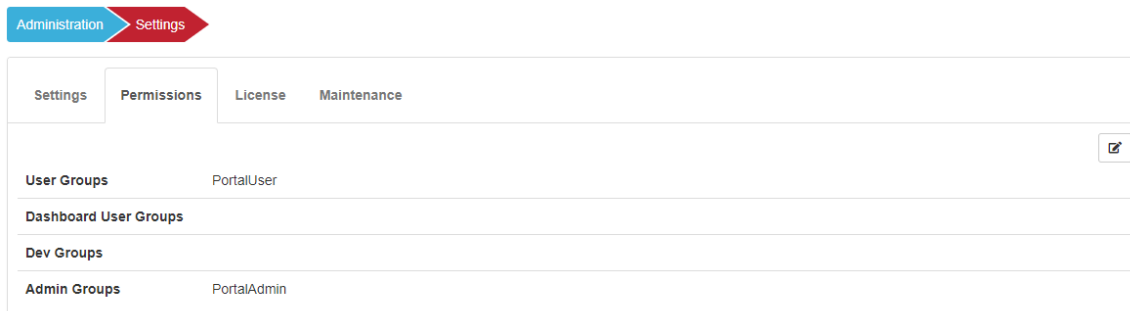
Administration Settings

Settings	License	Maintenance
Portal Name	Kelverion Demo Portal	
Portal Description	Welcome to the Kelverion Automation Portal	
Portal Logo		
Support Link	<a href="http://www.kelverion.com/contact">http://www.kelverion.com/contact</a>	
Enable Wiki	True	
Wiki Name	Wiki	
Query Timeout (sec)	60	
Currency Symbol	\$	
Stay On Request	False	
Enable Themes	True	
Azure Storage Container Name	portal-attachments	
Enable Reason For Rejecting Request	False	
Dashboard Update Interval (min)	5	

## Permissions

### On-Prem

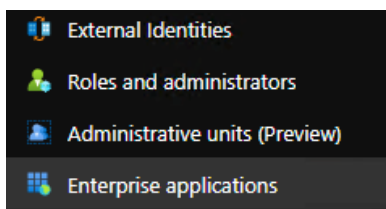
- **User Groups** – The Active Directory groups that have basic user permissions. Use the [Group Browser](#) to select groups. These AD User Group permissions include:
  - Viewing the portal **Services** view.
  - Viewing the portal **Requests** view.
- **Dashboard User Groups** – The Active Directory groups that have basic user permissions plus permission for the Dashboard. Use the [Group Browser](#) to select groups.
- **Dev Groups** – The active Directory groups that have access to build Services, Lists and Queries. Use the [Group Browser](#) to select groups.
- **Admin Groups** – The Active Directory groups that have admin user permissions. Use the [Group Browser](#) to select groups. These include in addition to all the AD User Group permissions:
  - Viewing the portal **Admin** view.



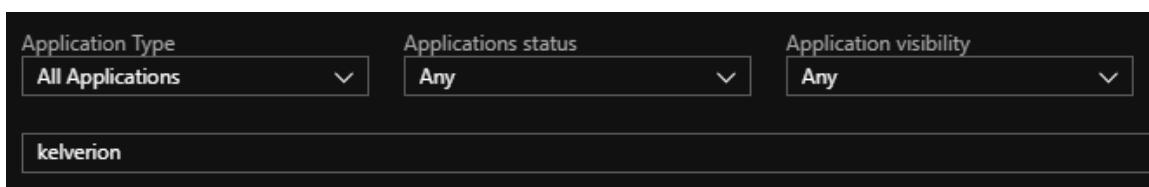
### Azure

This section is not available in the Azure version of the Automation Portal. User roles are administered in Azure Active Directory.

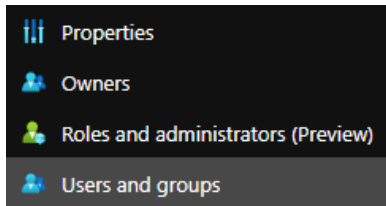
1. In the Azure Portal, select **Azure Active Directory**.
2. Select **Enterprise applications**.



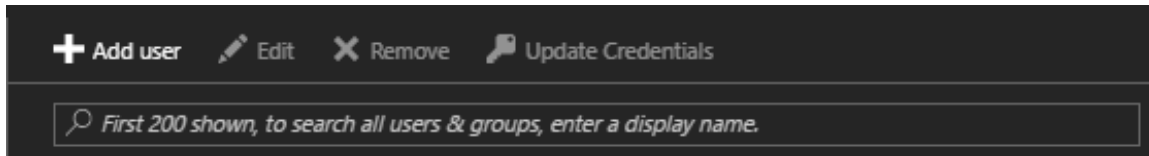
3. Select **All applications** and select the Automation Portal application. You also can search 'kelverion' and get the list of Kelverion applications.



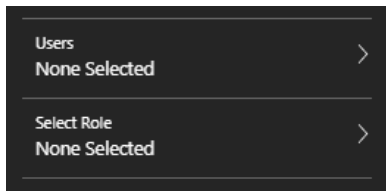
4. Select **Users and groups**.



5. Click **Add user**.

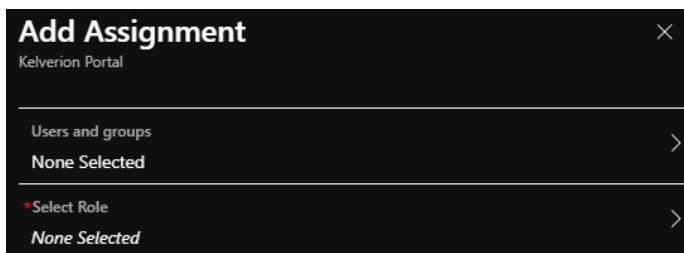


6. Select **Users**, assign user(s) to the application, and click **Select**. You can also select groups of users rather than individual users. Note: Nested Groups are not supported for roles in Azure currently.

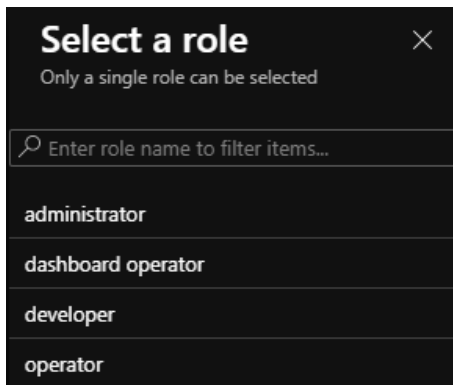


7. Select **Role** and assign a role to the selected user(s) or to a group with multiple users. The portal application offers the following roles and click **Select**.

- **Administrator** - Administrator can create Services and customize the portal.
- **Dashboard operator** – Dashboard users have basic user permissions plus permission for the Dashboard.
- **Developer** – Developer users have access to build Services, Lists and Queries.
- **Operator** - Basic operator and has no privileges to create new Services or customize the portal.



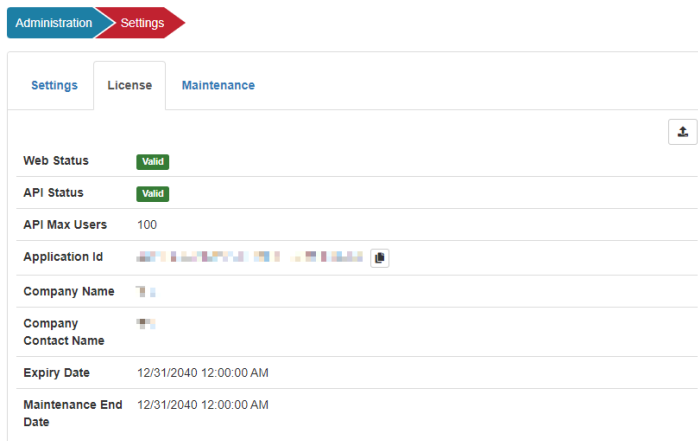




8. Click **Assign**.

## License

- **Web Status** – The status of the portal license: Invalid or Valid.
- **API Status** – Azure only - The status of the portal API license: Invalid or Valid.
- **API Max Users** – Azure only – Number of max API Users
- **FQDN** – On-Prem only - This is the FQDN of the server running IIS where the portal is hosted.
- **Application ID** – Azure only - This is the Application ID registered in the Active Directory domain. The Application Id can be found in the Details section.
- **Company Name** – The company name the license is registered to.
- **Company Contact Name** – The company contact name.
- **Expiry Date** – The date the license expires on.
- **Maintenance End Date** – The date the license expires on.



## Maintenance

The maintenance panel provides the following maintenance tasks:

- **Purge all deleted items** – This task permanently removes all portal records (Services, Offerings, Fields, Lists, Queries, Requests and Wiki) flagged as deleted.
- **Purge Log** – This task permanently removes all log records.
- **Purge Feedback** – This task permanently removes all feedback records.

Administration
Settings

Settings
Permissions
License
Maintenance

Purge all deleted items	
Purge Log	
Purge Feedback	

## Services

Using the Admin Services view you can add, modify, or delete services. Click **Add** to add a new service.

Administration
Services

Services
+

Add, modify or delete services.

No services found.

## Creating a Service

To create a new **Service**, specify the following and click **Add**.

- **Name** – The name of your service.
- **Description** – The description of your service.
- **Icon** – An icon to represent your service.
- **Color** – The color used for the service in the dashboard.
- **Active** – Determines if the service is visible in the Services view.

Administration
Services
New Service

Service Details

Name

Description

Icon

Color

#00aabb

Active

No
Yes

Add



## Service Details

To create a new **Service Offering**, you will use the **Service Details** view. From here you can edit your service by clicking **Edit**. Add a **Service Offering** by clicking **Add** in the **Service Offerings** panel.

Administration
Services
Your Service

Service Details

Your Service

Created	5/12/2021 12:57:10 PM
Updated	5/12/2021 12:57:10 PM
Owner	
Color	
Active	True

Service Offerings

Add, modify or delete an offering for this service.

No offerings found.

## Creating an Offering

To create a new **Offering**, specify the following and click **Add**.

- **Name** – The name of your offering.
- **Description** – describes your offering.
- **User Groups** – The Active Directory group that has permission to create this offering request and view their own requests. Use the [Group Browser](#) to select groups.
- **Team Groups** – The Active Directory group that has permission to create and view all offering requests created by all users. Use the [Group Browser](#) to select groups.
- **Requires External Ref. ID** – Adds an additional edit box on a request that can be used as an external reference ID.
- **Requires Group Approval** – Turns request approval on. When on the request will need to be approved by an approver in the specified group. Use the [Group Browser](#) to select groups.
- **AD Approval Group** – The Active directory group that has permission to approve this offering request.
- **Cost Saved** – The cost value this offering saves.
- **Time Saved** – The time value this offering saves.
- **Icon** – An icon to represent your offering.
- **Color** – The color used for the offering on the dashboard.
- **Tag** – A value added to the tag field of the request.
- **Active** – Determines if the offering is visible.

- **Mobile Active** – Azure only – Determines if the offering is visible in the Automation Portal mobile app. Offerings that are designed for mobile cannot contain File Attachment fields.
- **External URL** – A web URL that will be displayed as a link.

Administration > Services > Your Service > New Offering

### Offering Details

Name	<input type="text"/>
Description	<input type="text"/>
User Groups	<input type="text"/> +
Team Groups	<input type="text"/> +
Requires External Ref. ID	<input type="button" value="No"/> <input type="button" value="Yes"/>
Requires Group Approval	<input type="button" value="No"/> <input type="button" value="Yes"/>
Cost Saved	\$ <input type="text" value="0"/>
Time Saved	<input type="text" value="0"/> minutes
Icon	<input type="button" value="Icon"/>
Color	<input type="text" value="#00aabb"/> <input type="button" value="Color"/>
Active	<input type="button" value="No"/> <input checked="" type="button" value="Yes"/>
Mobile Active	<input type="button" value="No"/> <input type="button" value="Yes"/>
Tag	<input type="text"/>
External Url	<input type="text"/>

## Creating an Offering Folder

To create a new **Folder**, specify the following and click **Add**.

- **Name** – The name of your offering folder.
- **Description** – The description of your offering folder.
- **Active** – Determines if the offering folder is visible.
- **Mobile Active** – Azure only – Determines if the folder is visible in the Automation Portal mobile app.

Administration > Services > Your Service > New Folder

Folder Details

Name

Description

Active

Mobile Active

## Offering Details

If you just created a new **Offering**, you will now be in this view. Or you can return to this view by selecting an **Offering** in the **Service Details** view. From here you can edit your offering details by clicking **Edit** or you can add an **Offering Field** to the **Offering** by clicking **Add Field**.

Administration > Services > Your Service > Your Offering

Offering Details

Your offering description

Service

Your Service

Created

10/6/2021 2:33:07 PM

Updated

10/6/2021 2:36:12 PM

User Groups

Operator

Team Groups

Team

Requires External Ref. ID

False

Requires Group Approval

False

Owner

Color

Active

True

Cost Saved

\$0.00

Time Saved (min)

0

Tag

External Url

Markdown

Offering Fields

Add, modify or delete a field for this offering.

No fields found.

## Moving an Offering

You can move the offering from one service to another service offerings. Also, you have the option to move the offering to another offering folder.

On the edit offering page, you can use the option to move the offering from one service to another.

[Administration](#) > [Services](#) > [Your Service](#) > [Your Offering](#) > [Edit Offering](#)

### Offering Details

Name	<input type="text" value="Your Offering"/>
Description	<input type="text" value="Your Offering"/>
User Groups	<input type="text" value="Operator"/> +
Team Groups	<input type="text" value="Team"/> +
Requires External Ref. ID	<input type="radio"/> No <input type="radio"/> Yes
Requires Group Approval	<input type="radio"/> No <input type="radio"/> Yes
Cost Saved	<input type="text" value="\$ 0.00"/>
Time Saved	<input type="text" value="0"/> minutes
Icon	<input type="text" value="🔧"/>
Color	<input type="text" value="#00aabb"/> <input type="color" value="#00aabb"/>
Active	<input type="radio"/> No <input checked="" type="radio"/> Yes
Tag	<input type="text"/>
External Url	<input type="text"/>
Service	<input type="text" value="🔧 Your Service"/>
Folder	<div><input type="text" value="🔧 Your Service"/> <input type="text" value="🔧 Your Other Service"/> <input type="button" value="Save"/></div>

On the edit offering page, you can also move the offering to another offering folder.

Administration
Services
Your Service
Your Offering
Edit Offering

### Offering Details

Name

Description

User Groups

+

Team Groups

+

Requires External Ref. ID

No

Yes

Requires Group Approval

No

Yes

Cost Saved

\$

Time Saved

minutes

Icon

⚙

▼

Color

■

Active

No

Yes

Tag

External Url

Service

⚙ Your Service

▼

Folder

Your Service Root

▼

⚙ Your Service Root

📁 Your Folder

## Offering Markdown

Using the Offering Markdown, you can add a custom section to your offering that can be used to provide additional information to the user. When markdown is specified, it will appear on the right side of the Offering Request form. For more information on CommonMark click [here](#).

The Edit Markdown page contains the following:

- **Markdown Panel Title** – A custom title that appears on the Offering Request.
- **Text** – Your custom text can contain the following markdown using the tool bar to customize your text.
  - Bold
  - Italics
  - Heading
  - URL/Link
  - Image
  - Unordered List
  - Ordered List
  - Code
  - Quote

## Adding a Field

To add a new **Field** to an offering, specify the following and click **Add Field**.

- **Name** – Field name.
- **Help Text** – Helpful text located below the field.
- **Type** – The type of field.
  - **Date** – A date picker.
  - **Check Box** – A true of false selection.
  - **File Attachment** – A file attachment field allows the user to attach and upload a file with the submitted request.
    - **On-Prem** - The file is placed in the **Attachment Upload Folder** that is configure on the [Settings](#) page. The file is uploaded into a subfolder for the current month (ex. 072019 is July 2019) and each file is prefixed with the date and time the file was received. Maximum file attachment that can be uploaded is 50 MB.
    - **Azure** – The file is placed in the **Azure Storage Container** that is configure on the [Settings](#) page. The file is uploaded into a subfolder for the current month (ex. 072019 is July 2019) and each file is prefixed with the date and time the file was received. Maximum file attachment that can be uploaded is 50 MB. File Attachment fields should not be used in Offerings that have Mobile Active enabled.
  - **Hidden** – A hidden value that is not displayed to the user but is available for use by automation. You can specify your own hidden value, or you can specify **@User** for the username or **@ClientHostName** for the user machine name or IP address. **Note:** @ClientHostName is only available in the on-prem version of the portal.



- **List (Multiple Selection)** – A multiple selection drop down using a custom list or custom database table query. **Note:** When using a query with a list your query should be designed to return a suitable number of records for a drop-down list.
  - When configuring a query for a **List (Multiple Selection)** the query cannot define any output.
- **List (Single Selection)** – A single selection drop down using a custom list or custom database table query. **Note:** When using a query with a list your query should be designed to return a suitable number of records for a drop-down list.
- **Radio Button** – A single selection using a custom list.
- **Table (Display Only)** – A display only table. The results of this table are not included in the request.
  - When configuring a query for a **Table (Display Only)** the query cannot define any output.
- **Table (Multiple Selection)** – A multiple selection table using a custom database table query.
  - When configuring a query for a **Table (Multiple Selection)** the query cannot define any output.
- **Table (Single Selection)** – A single selection table using a custom database table query.
- **Text Area** – A multiple line text box.
- **Text Box** – A single line text box.
- **Time** – A time picker.
- **Secure Text Box** - A single line text box that is stored encrypted. For On-Prem [Always Encrypted Columns](#) must be enabled for the Secure Text Box option to appear.
 

**Note:** To build Orchestrator runbooks that will access and modify data stored in encrypted database columns, you must upgrade the Kelverion Integration Pack for SQL Server to version 3.2 or higher.

Similarly, when building Azure Automation runbooks that will access and modify data stored in encrypted database columns, you must upgrade to the Kelverion Integration Module for SQL Server version 2.4 or higher.
- **Default Value** – A default value that is prepopulated. (Text Box, Text Area, and Check Box only)
- **Regular Expression** – A regular expression used for validation. (Text Box and Text Area only)
  - **Options**
    - **Case insensitive** – Set regex to case insensitive.
- **Read Only** – Sets the field as read only and requires a Default Value. (Text Box and Text Area only)
- **Required** – Makes the field required for submission. (Text Box, Text Area, List, Date, Time, and Table only)
- **List** – A custom defined list. (Radio Button and List only)
- **Query** – A custom defined query. (Table only)
- **Use for Query Input** – Enables a **Text Box**, **List (Single Selection)** or **Radio Button** for use as a variable in Queries.

Administration > Services > Your Service > Your Offering > Add Field

### Field Details

Name

Help Text

Type

Default Value

Regular Expression

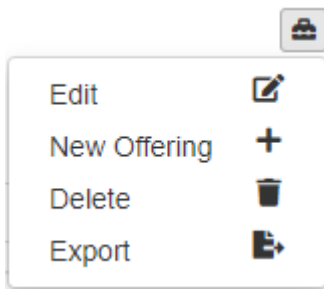
Use For Query Input

Read Only

Required

## Actions

Working with **Services**, **Offerings** and **Fields** you can use **Actions** to perform some common tasks. Actions can be selected from the Actions drop down menu.



### Services Actions

- **Edit** – Edit the Service details.
- **New Offering** – Add a new Offering to the Service.
- **Delete** – Remove the Service.
- **Export** – Export the Service.
- **Move to Top** – Move the Service to the top.
- **Move Up** – Move up the Service.
- **Move Down** – Move down the Service.
- **Move to Bottom** – Move the Service to the bottom.

### Offering Actions

- **Edit** – Edit the Offering details.
- **Delete** – Remove the Offering.
- **Export** – Export the Offering.
- **Move to Top** – Move the Offering to the top.

- **Move Up** – Move up the Offering.
- **Move Down** – Move down the Offering.
- **Move to Bottom** – Move the Offering to the bottom.

### Field Actions

- **Edit** – Edit the Field details.
- **Delete** – Remove the Field.
- **Export** – Export the Field.
- **Move to Top** – Move the Field to the top.
- **Move Up** – Move up the Field.
- **Move Down** – Move down the Field.
- **Move to Bottom** – Move the Field to the bottom.

## Lists

Using the Admin Lists view you can add, modify, or delete lists. Click **New List** to add a new list.

The screenshot shows the 'Administration > Lists' breadcrumb trail. Below it, there's a header 'Lists' with a '+' icon and a trash icon. The main content area says 'Add, modify or delete lists.' followed by a search icon and the message 'No lists found.'

## Creating a List

To create a new **List**, specify the following and click **Action -> Add List**.

- **Name** – The name of your list.
- **Description** – The description of your list.
- **Type** – The type of list **Select List** or **Radio Button**.

The screenshot shows the 'List Details' form. It has a breadcrumb trail 'Administration > Lists > New List'. The form contains three input fields: 'Name', 'Description', and 'Type' (which is a dropdown menu with 'Select List' as the current selection). Below these fields is a blue 'Add' button.

## List Details

Using the **List Details** view you can manage your list. To edit your **List** details, click **Edit**. To add new values to your list, click the **New List Value**.

Administration
Lists
Your List

### List Details

Your List

Type	Select List
Created	5/12/2021 1:20:15 PM
Updated	5/12/2021 1:20:15 PM

### List Values

Add, modify or delete a list value.

+

No list values found.

## Creating a List Value

To create a new **List Value**, specify the following and click Add.

- **Value** – The value to add.
- **Description** – The description of your list value.

Administration
Lists
Your List
New List Value

### List Value Details

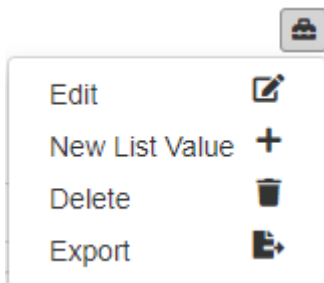
Value

Description

Add

## Actions

Working with **Lists** and **List Values** you can use **Actions** to perform common tasks. Actions can be selected from the Actions drop down menu.



### List Actions

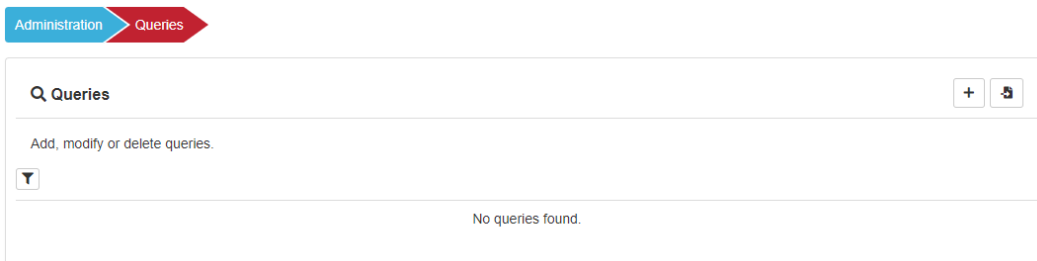
- **Edit** – Edit the List details.
- **New List Value** – Add a new List Value to the List.
- **Delete** – Remove the List.
- **Export** – Export the List.

### List Value Actions

- **Edit** – Edit the List Value details.
- **Delete** – Remove the List Value.
- **Export** – Export the List Value.
- **Move to Top** – Move the List Value to the top.
- **Move Up** – Move up the List Value.
- **Move Down** – Move down the List Value.
- **Move to Bottom** – Move the List Value to the bottom.

## Queries

Using the **Admin Queries** view you can add, modify, or delete queries. The Query function can be used to Select data from a SQL Server Database to present as a pick list to a User through the Automation Portal. Click **New Query** to add a new query.



## Security Recommendations

The Queries capability of the Automation Portal enables you to compose your own SQL Queries for extracting the data you require, to display in the Portal, from Databases within your environment, or to execute SQL Stored Procedures that will extract the data you require.

To enable the execution of SQL Queries or Stored Procedures, you must provide a User Account in SQL Server with the necessary permissions to execute the required SQL commands. To minimize any security risks associated with running SQL Queries from within the Automation Portal and to avoid any abuse of the system access we recommend that you:

- Create a special “Least Privilege” DB account for use with the Automation Portal Query and Stored Procedure features.
- For SQL Queries, use a database user with Select-only permissions to the required tables.
- Alternatively, use Stored Procedures exclusively and limit the database user permissions to only those Stored Procedures required for Automation Portal operation.

Guidance on SQL Permissions can be found at: <https://docs.microsoft.com/en-us/sql/relational-databases/security/permissions-database-engine?view=sql-server-ver15>

## Creating a Query

To create a new **Query**, specify the following and click **Add**.

- **Name** – The name of your query.
- **Description** – The description of your query.
- **Database Server** – The name or IP address of your database server.
- **Database Name** – The name of the database.
- **Authentication Type** – **Windows Authentication** or **SQL Authentication**.
  - **Windows Authentication** – When using Windows Authentication, the Automation Portal IIS app pool user must have the appropriate access to any database you wish to access using a query.
- **Login** – Username for SQL Authentication.
- **Password** – Password for SQL Authentication.
- **Parameters**
  - **Query Parameters** – Assign an output from another Query to a variable that can be used in the SQL Query.
  - **Client Parameters** – Turns on @User or @ClientHostName variables.
    - **@User** – contains logged on user's name.
    - **@ClientHostName** – contains the logged-on user's client host name or IP address.
  - **Field Parameters** – Assign an output from a Text field or Single Select list to a variable that can be used in the SQL Query.
- **Command Type** – SQL Query or Stored Procedure.
- **SQL Query** – SQL query.
  - **Note:** "Order By" is not supported.
- **Stored Procedure** – Stored procedure name.
  - **Note:** "Order By" is not supported.
- **Outputs** – The columns that you wish to define as output available to other Queries.

**Note:** The specified SQL Query or Store Procedure do not support tables with columns with spaces.

### Q Query Details

Name

Description

Database Server

Database Name

Authentication Type

Windows Authentication

Parameters

Query Parameters

Client Parameters

Field Parameters

Parameter Name

@

Source Query

Select Source Query

Source Query Output

Select Source Query Output

Add

Reset

Parameter Name	Source Query	Source Query Output
No matching records found		

Command Type

SQL Query

SQL Query

Test

Outputs

Output Name

Add

Reset

Output Name
No matching records found

Add

**Note:** See [Security Considerations](#) for additional information on securing your data.

## Parameters and Outputs

Using **Parameters** and **Outputs** in a Query you can link Queries together to build cascading field selections and build dynamic Offerings. To start using **Parameters** and **Outputs**, you need to start with defining at least one **Query** that has **Outputs**. Queries can specify multiple outputs and define multiple variables.

### Outputs

**Outputs** determine which columns of the **SQL Query** will be available as **Source Query Output** in another query.

**Output Names** must match a column name in the **SQL Query**. For example, if the query is

***SELECT Name, ProductCategoryID FROM Production.ProductCategory***

then valid outputs would be **Name** and/or **ProductCategoryID**.

To add **Outputs**, enter the **Output Name** and click **Add**.

**Outputs**

Output Name

Output Name
No matching records found

## Parameters

### Query Parameters

A **Query Parameter** can be defined as an output from a **Source Query**. The variable can then be used in the **SQL Query** or **Stored Procedure**. See [Outputs](#) for defining an output.

To add **Query Parameters**, enter a unique **Parameter Name**, select a **Source Query** and a **Source Query Output** you wish to assign to the variable and click **Add**.

**Parameters**

Query Parameters Client Parameters Field Parameters

Parameter Name

Source Query

Source Query Output

Parameter Name	Source Query	Source Query Output
No matching records found		

### Client Parameters

Client Parameters are special parameters that define information about the current user. Two client parameters are available. The following Client Parameters can be activated.

- @User – Username of the logged in user




- @ClientHostName – Computer name or IP address of the logged in user's computer
  - **Note:** @ClientHostName is not supported in the Azure version of the portal.


**Parameters**

Query Parameters
 Client Parameters
 Field Parameters

**@User** ☐ No ☒ Yes

E.g.: @User = 

**@ClientHostName** ☐ No ☒ Yes

E.g.: @ClientHostName = 

### @User Parameter

The Portal has a special parameter called @User which allows you to find out which User is currently logged into to the Portal. You can use this variable as part of any query where you want to limit the data displayed to information which is specific to the User who is currently logged into the Portal i.e., VMs I Own, Active Directory Groups I am a member of. The variable will return you the Active Directory UserPrincipalName of the person logged into the Portal in the format:

On Premise Active Directory	Domain\Username
Azure Active Directory	<a href="#">username@company.com</a>

The @User variable is a built-in function of the Portal and therefore you do not need to pre-define it before you can use it in a Query.

For example, if you want to show the VMs you own, you will use the variable in the query like this:

```
SELECT VMName FROM Production.Assets WHERE Owner = @User
```

When specifying the variable in the **SQL Query** you must include the @ symbol.

### @ClientHostName parameter

**Note:** This option is only available in the on-prem version of the portal.

The Portal has a special parameter called @ClientHostName which allows you to find out the client host name or IP address of the logged in user.

If the computer name cannot be determined, the IP address is returned.

When specifying the variable in the **SQL Query** you must include the @ symbol.

Please note that @ClientHostName functionality is only available for the Automation Portal on the on-prem version of the Automation Portal.

## Field Parameters

A **Field Parameter** can be defined as a **Text Field** or **List (Single Selection)** to the parameter. The parameter can then be used in the **SQL Query**. To enable a **Text Field** or **List (Single Selection)** to support variables, the field must have **Use for Query Input** enabled. See [Adding a Fields](#) for enabling a Text Field for variable use. See [Outputs](#) for defining an output.

To add **Field Parameters**, enter a unique **Parameter Name** and select a **Field Name** you wish to assign to the variable and click **Add**.

**Parameters**

Query Parameters
Client Parameters
Field Parameters

Parameter Name

Field Name
Select Field Name

Add
Reset

Parameter Name	Text Field Name
No matching records found	

## Using Parameters

### SQL Query

To use **Parameters** in a **SQL Query**, you specify the parameter name in your **SQL Query**. For example, if you have defined a parameter named **@ProductCategoryID**, you would use the parameter in the query like this:

```
SELECT ProductSubcategoryID, Name FROM Production.ProductSubcategory WHERE ProductCategoryID = @ProductCategoryID
```

When specifying the parameter in the **SQL Query** you must include the @ symbol.

### Store Procedure

To use **Parameters** in a **Stored Procedure**, you specify the parameter name in your **Stored Procedure**. For example, if you have defined a parameter named **@ProductCategoryID**, you would use the parameter in the procedure like this:

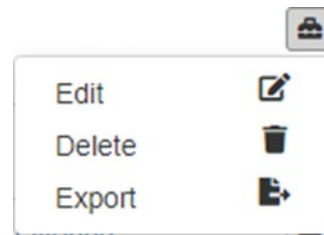
```
CREATE PROCEDURE [dbo].[GetProductSubcategory] (@ProductCategoryID int) AS Select ProductSubcategoryID from Production.ProductSubcategory where ProductCategoryID = ProductCategoryID
```

## Actions

Working with **Queries** you can click **Actions** to perform common tasks. Actions can be selected from the Actions drop down menu.

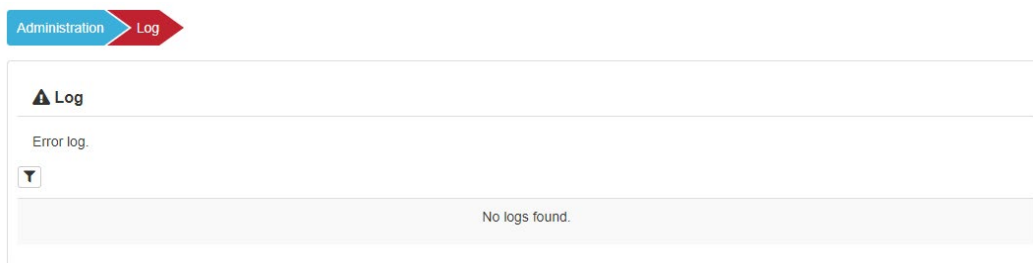
## Query Actions

- **Edit** – Edit the Query details.
- **Delete** – Remove the Query.
- **Export** – Export the Query.



## Log

The Log located on the **Admin** page can be used to diagnose any errors that have occurred during the use of the Automation Portal and can only be accessed by an Administrator of the Automation Portal.



## Log Details

Clicking on a Log entry will display the Log Details page which provides information on the error that has occurred.

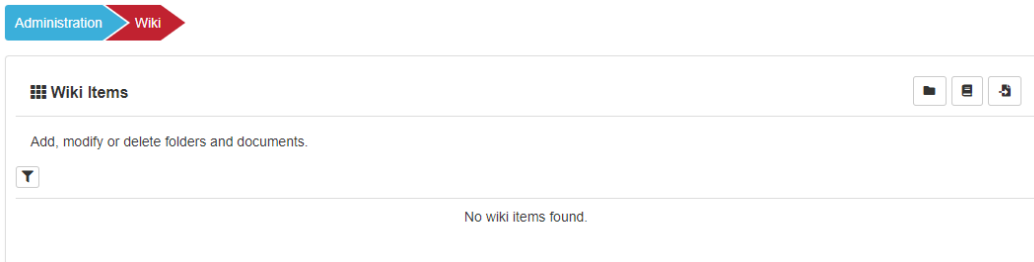
- **Id** – Unique Id of the Log in the portal database
- **Create** – The date the error was logged.
- **User** – The user that was using the portal when the error occurred.
- **URL** – The URL that was called when the error occurred.
- **Controller Name** – Controller the error occurred in.
- **Action** – The action in the controller the error occurred in.
- **Exception Message** – The error message of the error that has occurred.
- **Exception Stack Trace** – Detail stack trace of the error that has occurred.

### Log Details

<b>Id</b>
<b>Created</b>
<b>User</b>
<b>Url</b>
<b>Controller Name</b>
<b>Action</b>
<b>Exception Type</b>
<b>Exception Message</b>
<b>Exception Stack Trace</b>

## Wiki

Using the Wiki view you can add, modify, or wiki folders and documents. Click **New Folder** or **New Doc**.



## Creating a Folder

To create a new **Folder**, click **New Folder** and specify the following.

- **Name** – The name of your Wiki folder.
- **Description** – The description of your Wiki folder.
- **Users Group** – The AD group permitted to view the Wiki folder. Use the [Group Browser](#) to select groups.
- **Active** – Determines if the folder is visible in the Wiki view.

## Creating a Doc

To create a new **Doc**, click **New Doc** and specify the following.

- **Name** – The name of your doc.
- **Description** – The description of your doc.
- **Users Group** – The AD group permitted to view the doc. Use the [Group Browser](#) to select groups.
- **Active** – Determines if the doc is visible in the Wiki view.
- **Text** – The documents content. For more information on CommonMark click [here](#).

Administration > Wiki > New Doc

### Document Details

Name

Description

User Groups  +

Active ☐ No ☒ Yes

Text

**B I H** Preview

## Filters

The admin tables can be filtered by clicking on the filter builder icon at the top of each admin table.

AND OR + Add rule + Add group

No filters specified ✕ Delete

## Import/Export

### Exporting

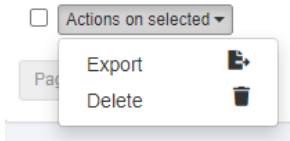
#### Export All

Click **Export All** in the **Services**, **Offerings**, **Fields**, **Lists**, **List Values**, **Queries** and **Wiki Items** pages, to export the data in each of these sections, respectively. When at the top level of Service, the export will include all items that make up the Service. This concept applies to **Offerings**, **Fields**, and **Lists**.


#### Export Selection

Individual components can be exported using the check box next to each component and clicking **Actions on selected** and the **Export** option.

**Note:** Components are exported in the order they are checked and when imported will appear in the checked order.



## Import

Click **Import**  available in the **Services, Offerings, Fields, Lists, List Values** and **Queries**, you can import data into each of these sections. When importing if any duplicate names are found in the database, a confirmation prompt will be presented with the option to cancel or import.

## Group Browser

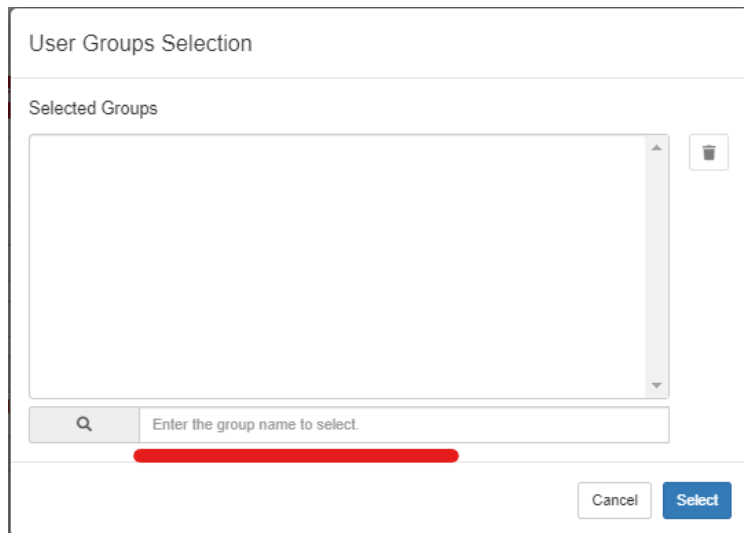
The group browser is used throughout the Portal for user group selection. For Role-based Access Control (RBAC) operations in the Portal, only valid Windows Active Directory or Azure Active Directory groups can be used. The Group Browser allows you to find Groups which already exist and select them for RBAC use. The following portal areas have group browsers.

- [Admin -> Settings -> Permissions \(On-Prem\)](#)
  - User Groups
  - Dashboard User Groups
  - Dev Groups
  - Admin Groups
- [Offering](#)
  - User Groups
  - Team Groups
  - Approval Groups
- [Wiki Doc](#)
  - User Groups
- [Wiki Folder](#)
  - User Groups

1. Click the plus sign next to the Group to bring up the Group Browser

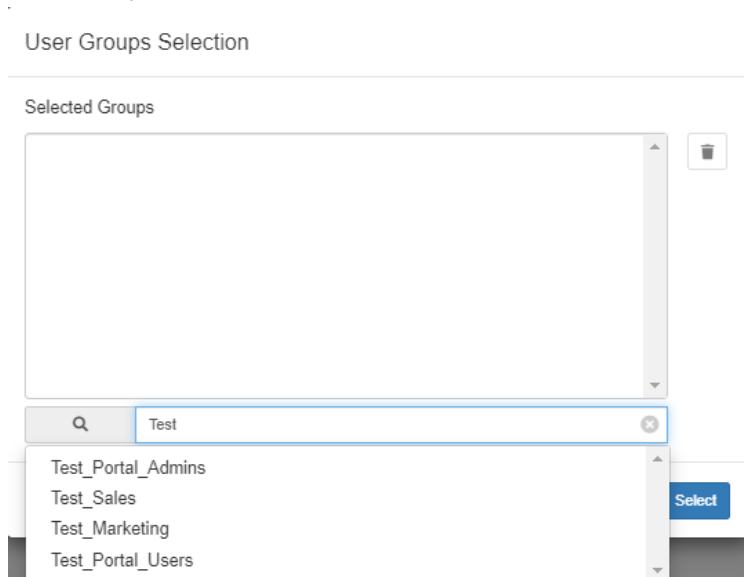
A screenshot of a form with two input fields. The first field is labeled 'User Groups' and has a plus sign in a small box to its right. Below this field, there is a red error message: 'The User Groups field is required.' The second field is labeled 'Team Groups' and also has a plus sign in a small box to its right.

2. The User Groups Selection browser window appears. In the search box, where it says, **Enter the group name to select**,



The 'User Groups Selection' dialog box features a 'Selected Groups' section with a large empty list box and a trash icon. Below this is a search bar with a magnifying glass icon and the placeholder text 'Enter the group name to select.'. A red horizontal bar is positioned below the search bar. At the bottom right, there are 'Cancel' and 'Select' buttons.

start typing the name of the Group you want to use. The browser will start to query Active Directory/Azure Active Directory and display a list of Groups whose names start with the text you entered, the more precise your name entry the less results will be shown.



This screenshot shows the 'User Groups Selection' dialog box with the search bar containing the text 'Test'. A dropdown menu is open below the search bar, displaying a list of group names: 'Test\_Portal\_Admins', 'Test\_Sales', 'Test\_Marketing', and 'Test\_Portal\_Users'. The 'Select' button is visible at the bottom right.

- Click on one of the Groups shown and it will be added to the **Selected Groups** area. Repeat the search and select to add multiple Groups.

The screenshot shows a dialog box titled "User Groups Selection". Inside, there is a section labeled "Selected Groups" which contains a list box with the text "Test\_Portal\_Users". To the right of the list box is a trash icon. Below the list box is a search bar with a magnifying glass icon and the placeholder text "Enter the group name to select.". At the bottom right of the dialog box are two buttons: "Cancel" and "Select".

- Once you have selected all the groups click **Select** to save your selection. To remove a group, select the group in the list and click the trash button.

## Markdown Syntax

Markdown can be used in the following locations:

- Offering Markdown
- Wiki Doc
- In the Message field of a Request.

For more information on CommonMark click [here](#).

## Azure

### Client Secret

The Automation Portal uses an Azure Application Client Secret which has an expiry date and will need to be updated before the expiry date.

### Updating Client Secret

To update the client secret.

- Navigate in the Azure Portal to **Azure Active Directory -> App Registration -> <your app registration> -> Client secrets**.
- Click **New client secret**.
- Enter values for **Description** and **Expires**.
- Click **Add**.
- Copy the new client secret value.



6. Navigate in the Azure Portal to **Resource Groups** -> <your automation portal group> -> <your automation portal app service> -> **Settings** -> **Configuration** -> **Application settings**.
7. Click **AzureAd:ClientSecret**.
8. Update the **Value** with the copied secret value from step 5.
9. Click **OK**.
10. Click **Save**.

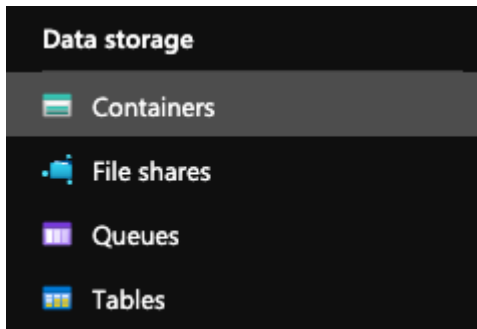
## Storage

An Azure Storage account and container is required for the File Attachment field when using the Azure version of the Automation Portal.

### Container Name

The container name can be in your Azure storage account.

1. Locate an existing storage account or [create a storage account](#).
2. Locate the section **Data Storage**.



3. Select **Containers**.
4. Locate the **name** of an existing container or create a new one.

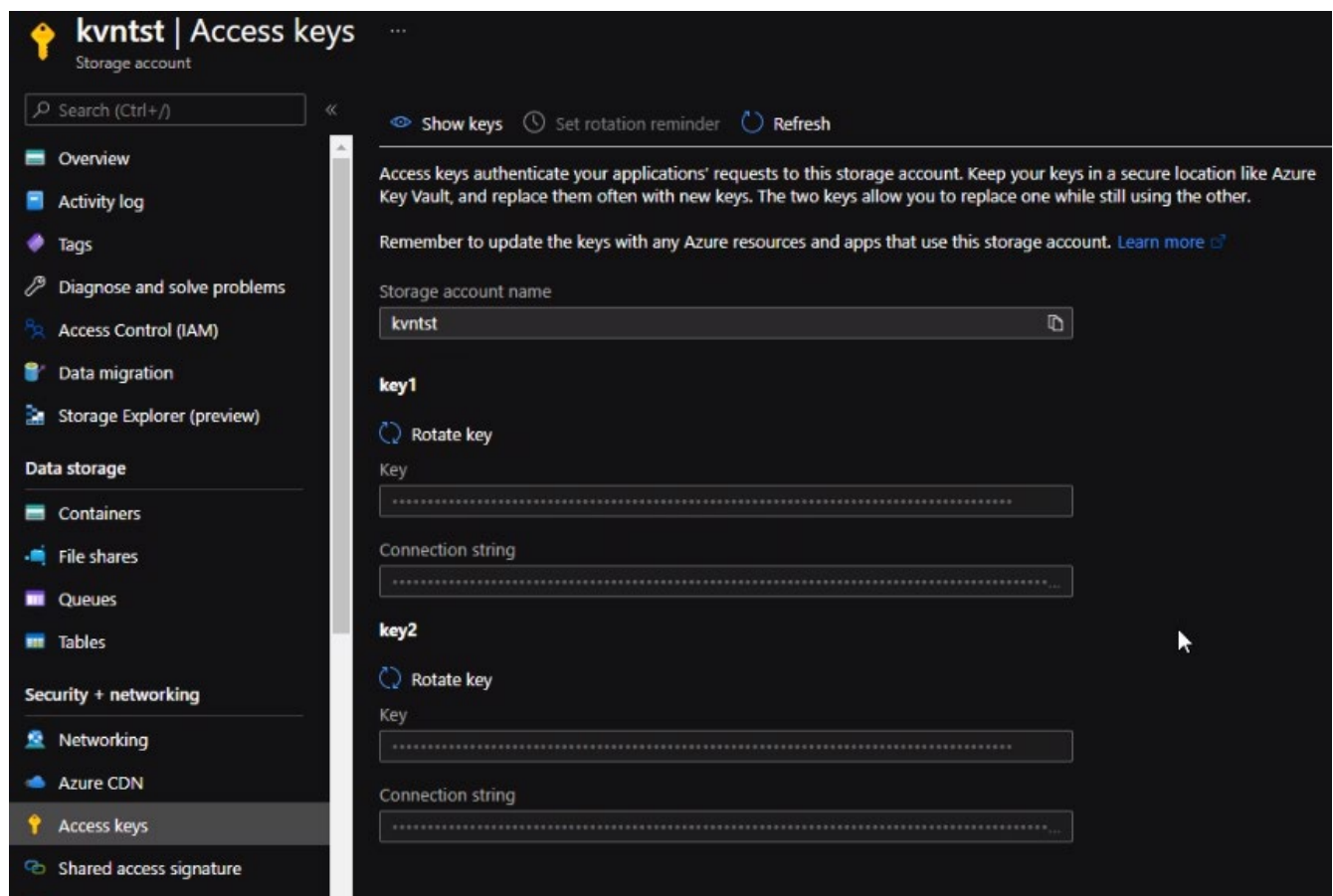
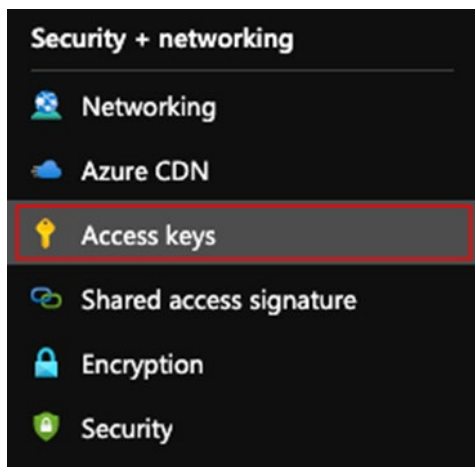


- Record the Name of the Container for entering the Portal Settings.

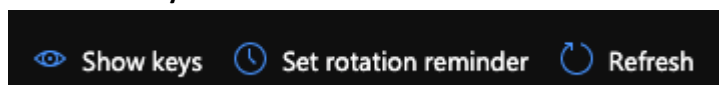
### Connection String

The Azure storage account connection string can be in Azure.

1. In the Azure Portal navigate to the Storage Account view.
2. Locate an existing storage account or [create a storage account](#).
3. Locate the **section Security + networking section**.
4. Select **Access Keys**.



5. Click **Show Keys**.



6. Copy one of the **connection strings** for your environment from key1 or key2. Contact your administrator for assistance on which key to use. Both keys will work but one key preferred over the other.

- Go to the Portal Settings screen and enter the Container Name and Connection string information.

Administration > Settings > Edit Settings

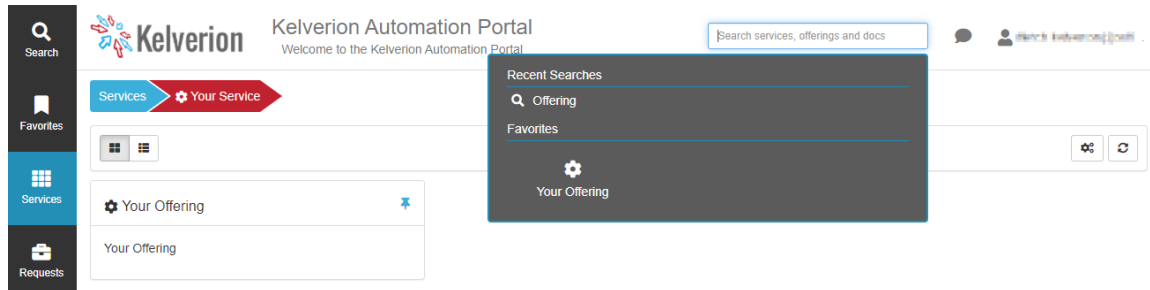
### Settings

Portal Name	<input type="text" value="Kelverion Demo Portal"/>
Portal Description	<input type="text" value="Welcome to the Kelverion Automation Portal"/>
Support Link	<input type="text" value="http://www.kelverion.com/contact"/>
Enable Wiki	<input type="radio"/> No <input checked="" type="radio"/> Yes
Wiki Name	<input type="text" value="Wiki"/>
Query Timeout (sec)	<input type="text" value="60"/>
Currency Symbol	<input type="text" value="\$"/>
Stay On Request	<input type="radio"/> No <input type="radio"/> Yes
Enable Themes	<input type="radio"/> No <input checked="" type="radio"/> Yes
Azure Storage Account Connection String	<input type="text" value="DefaultEndpointsProtocol=https;AccountName=kvtst;AccountKey=ZVXg;8w==;EndpointSuffix=core.windows.net"/>
Azure Storage Container Name	<input type="text" value="portal-attachments"/>
Enable Reason For Rejecting Request	<input type="radio"/> No <input type="radio"/> Yes
Dashboard Update Interval (min)	<input type="text" value="5"/>

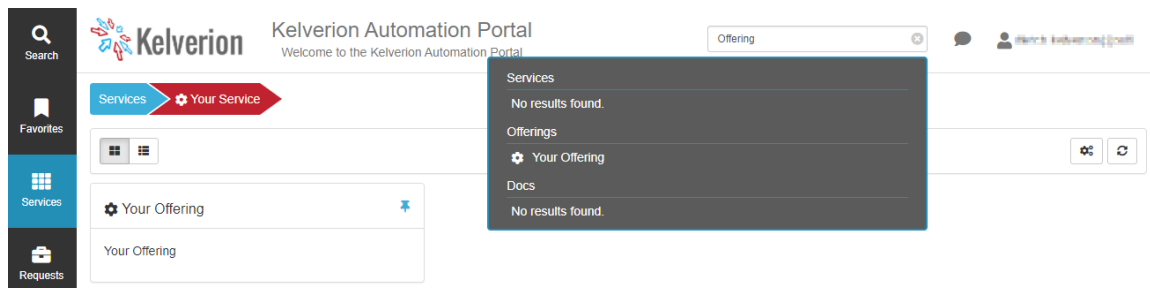
## Global Search

The search box in the portal top menu provides a search across Services, Offerings and Docs. Searching both the name and description fields. The search result box also provides quick access to your favorite offerings. Search is case insensitive.

- **Recent Searches** – Your last five searches.
- **Favorites** – The 12 most recent pinned offerings.



- **Services** – The services that match your search.
- **Offerings** – The offerings that match your search.
- **Docs** – The docs that match your search.



## Search View

The search view can be used to search **Services**, **Offerings** and **Docs**. Search is case insensitive.

Fields that are searched:

- Name
- Description

Kelverion Automation Portal  
Welcome to the Kelverion Automation Portal

Search services, offerings and docs

Search

Search services, offerings and docs

Services (1) Offerings (1) Docs (0)

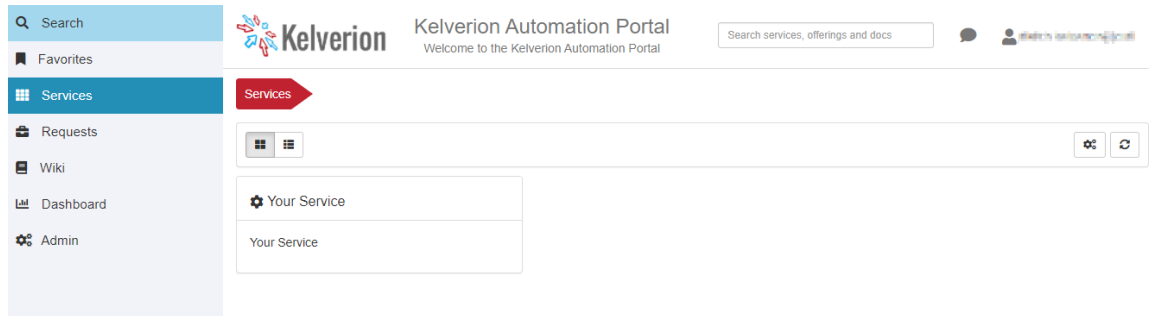
Name	Description	Created
Your Service	Your Service	5/12/2021 4:57:10 PM

Page 1 of 1 1

Showing 1 to 1 of 1 rows 10 rows per page

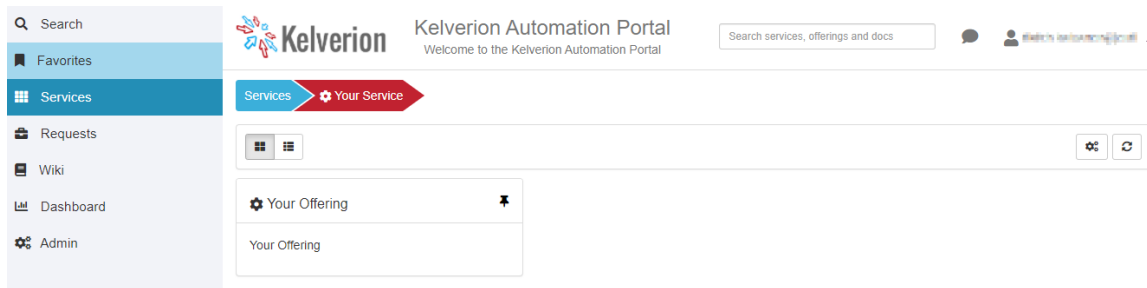
## Services View

The Services view displays each **Service** as a panel under the heading **Services**. To view the offerings in a Service, click the name of the Service to enter the Offering view for that Service.



## Service Offerings

The **Offering** view lists all the Offerings as a panel for the Service you selected. Click the name of the Offering to create a Request. If the Administrator has added an External URL, the description link will open a new browser window using that link.



## New Request

The **New Request** view lists all the Fields defined for the offering. This can be customized using the [Admin Services](#) view. When the user clicks the **Submit**, any fields that have validators will be validated and any validation errors will be display below each field. If optional [markdown](#) is specified, it will appear to the right of the request fields.

The screenshot displays the Kelverion Automation Portal interface. On the left is a sidebar with navigation links: Search, Favorites, Services (highlighted), Requests, Wiki, Dashboard, and Admin. The main header area includes the Kelverion logo, the title 'Kelverion Automation Portal', a welcome message, a search bar, and user avatars. Below the header is a breadcrumb trail: Services > Your Service > Your Offering - New Request. The main content area is divided into two panels. The left panel, titled 'Fields', contains a legend for field types (Required field, Optional field, Display only) and a form section for 'Your Text Box Field' with a text input and a 'Submit' button. The right panel, titled 'Work Instructions', contains a block of placeholder Lorem Ipsum text.

## Request Details

After submitting a request, the Request Details view will be displayed. The Request Details view lists the details of the newly created Request.

- **ID** – The unique identifier of the request.
- **Service** – The root Service that contains the Offering.
- **Offering** – The Offering type of this Request.
- **Created** – The date and time the Request was created.
- **Updated** – The date and time the Request was updated.
- **State** – The current State the Request is in.
- **Requested By** – The user that created the request.
- **Runbook Owner** – A field used to identify the automation runbook that processed the request.
- **Message** – A field used for messages from the automation runbook that processed the request. The message field supports text in markdown format. For more information on CommonMark click [here](#).
- **Pre-fill URL** - It will be displayed for the user with Admin rights. The Pre-fill URL is only displayed for when the query string portion of the URL does not exceed the [maxQueryStringLength](#) defined in the web.config file.
- **File attachment** - It will display the server location where the attachment is saved.

The screenshot displays the Kolverion Automation Portal interface. On the left is a sidebar with navigation links: Search, Favorites, Services, Requests (highlighted), Wiki, Dashboard, and Admin. The main header area includes the Kolverion logo, the portal name, a welcome message, a search bar, and user profile icons. Below the header, a red banner indicates 'Requests' and 'Request Id: 184'. The central content area is divided into three sections: 'Details', 'Fields', and 'History'. The 'Details' section contains a table with request information. The 'Fields' section shows a single text box field. The 'History' section shows a single entry for the request's creation.

Details	
Id	184
Service	<a href="#">Your Service</a>
Offering	<a href="#">Your Offering</a>
Created	5/12/2021 1:41:41 PM
Updated	5/12/2021 1:41:41 PM
State	New
Requested By	<a href="#">Mark Kolverion@portal.com</a>
Runbook Owner	
Message	
Cost Saved	\$0.00
Time Saved (min)	0
Tag	
External Ref. ID	
Offering Pre-Fill URL	<a href="#">https://portalUrl.com/Offering/NewRequest?offeringId=184&amp;fieldid=23527=Test</a>

Fields	
Your Text Box Field	Test

History	
5/12/2021 1:41:41 PM	Created <a href="#">Mark Kolverion@portal.com</a>

## Stay On Request

An optional setting is available in the [Admin Settings](#) view that allows the **Offering Request** to stay on the **New Request** view. Turning **Stay on Request** on will keep the view on the **New Request** view after submission.

Query Timeout	60
Stay On Request	True

## Offering Pre-Fill URL

The Offering Pre-Fill URL is viewable by an administrator and provides a reusable link to populate the offering form with the values specified in the pre-fill URL query string. The Pre-fill URL is only displayed for when the query string portion of the URL does not exceed the [maxQueryStringLength](#) defined in the web.config file.

### Pre-Fill URL Syntax

Replace **<offeringId>** with the Id of the offering you wish to create. Define one or more fieldid=value pairs with the fields you wish to set.

`https://portalUrl.com/Offering/NewRequest/?offeringId=<offeringid>&fieldid=value&fieldid=value`

### Field Syntax

**Tip:** Use the Offering Pre-Fill for an existing request as a starting point for creating your own custom URL.



ExternalId	Specify a value of external id
Text Box	Specify a value of text box
Text Area	Specify a value of text area
Check Box	Specify True or False
Radio Button	Specify a valid radio button value
List (Single Selection)	Specify a valid list value
List (Multiple Selection)	Specify valid list values in comma separated value
Date	Specify date in YYYY-MM-DD (ex. 2019-07-16)
Time	Specify time in hh:mm AM (ex. 10:45 AM)
Table (Single Selection)	<p>Specify table selection in the following JSON format. One or more columns can be specified. Replace <b>&lt;columnName&gt;</b> with your column name and <b>&lt;value&gt;</b> with the value you wish to set. Remember your values must match a valid value for the table.</p> <pre>{   "table": {     "row": {       "&lt;columnName&gt;": "&lt;value&gt;"     }   } }</pre>
Table (Multiple Selection)	<p>Specify table selection in the following JSON format. One or more columns can be specified. Replace <b>&lt;columnName&gt;</b> with your column name and <b>&lt;value&gt;</b> with the value you wish to set. Remember your values must match a valid value for the table.</p> <pre>{   "table": {     "row": [       {         "&lt;columnName&gt;": "&lt;value&gt;"       },       {         "&lt;columnName&gt;": "&lt;value&gt;"       }     ]   } }</pre>

### Querystring Length

The length of your Pre-Fill URL may exceed the default values of the web server. These can be increased in the web.config file by locating and increasing the following values.

- **<httpRuntime>**
  - **maxUrlLength** specifies the length of the URL in number of characters. The default is 260.
  - **maxQueryStringLength** specifies the length of the query string in number of characters. The default is 2048.
- [<requestLimits>](#)

- **maxQueryString** specifies the maximum length of the query string in bytes. The default value is 2048.
- **maxUrl** – specifies the maximum length of the URL in bytes. The default value is 4096.

## Requests View

The **Request** view gives you a view of all the Requests you have created. Using the **Request** view, you can see the following information for each request. Several Actions are available allowing quick access to common tasks.

## Views

- **My Requests** – Displays the requests of the current user.
- **All Requests** – Displays all requests for all Offerings the current user is authorized for.
- **Pending Approval** – Displays all requests waiting for the current user's approval.

## Query Builder

Queries are helpful in locating requests. Build a query using the **Add rule** and **Add group** buttons.

## Building a Query

1. Click **Add rule**.

2. Select field to filter. Available options are ID, Service, Offering, Created Date, Updated Date, State, Requested By, External Ref. ID.
3. Select filter operation. Available options depend on the selected field.
4. Specify the filter value.
5. Click **Apply Query**.

**Note:** Created and Updated Date filters will only return Requests created with Automation Portal 2.2 or higher.

## Saving a Query

Queries can be saved for later use. To save a query:

1. Defined your query.
2. Click **Save Query**.
3. In the Save Query dialog specify a Query name.
4. Click **Save**.

## Using a Saved Query

Saved queries can be selected from the Saved Queries drop down. Once selected the query will be run.

## Request Details

- **ID** – The unique identifier of the request.
- **Service** – The root Service that contains the Offering.
- **Offering** – The Offering type of this Request.
- **Created** – The date and time the Request was created.
- **Updated** – The date and time the Request was updated.
- **State** – The current State the Request is in.
- **Requested By** – The user that created the request.
- **External Ref. ID** – A field that can be used for External Reference ID.
- **History** – A history of actions performed on the Request.

## Request Fields

Request Fields can be viewed by clicking on the ID of the Request to view the Request in the Request Details view. Or you can click the + symbol for an expanded view that will list all the fields in the Request.

ID	Service	Offering
+	1	Your Service Your Offering

Figure 1 - Clicking the + expands the field details.

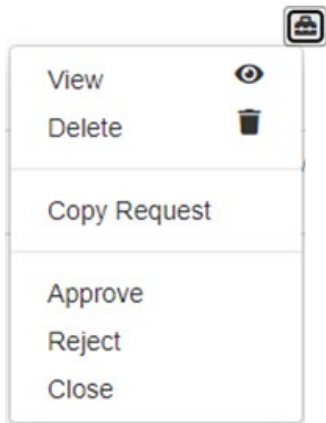
<input type="checkbox"/>	ID	Service	Offering	Created	Updated	State
<input type="checkbox"/>	1	Your Service	Your Offering	4/8/2019 1:46:46 PM	4/8/2019 1:46:46 PM	New

Your Text Box Field	Your Value
---------------------	------------

## Request Actions

Working with **Requests** you can use **Toolbox icon** to perform some common tasks.



Actions can be selected once the user clicks on the gear drop down menu.

- **View** – View the Request details.
- **Delete** – Remove the Request.
- **Copy Request** – A quick way to resubmit the request with previous submitted values.
- **Approve** – Set the state to approved.
- **Reject** – Set the state to rejected.
- **Close** – Set the state closed.

## Multi-Requests Actions

You can select **Multiple Requests** and provide the same common tasks. The tasks can be different depends on the different AD Security groups. Actions can be selected from the Actions drop down menu.

- **Delete** – Remove the Request.
- **Approve** – Set the state to approved.
- **Reject** – Set the state to rejected.
- **Close** – Set the state closed.

ID	Service	Offering	Created	Updated	State	Requested By
5210	Your Service	Your Offering	11/25/2020 12:22:22 PM	11/25/2020 12:22:22 PM	New	[User Avatar]

Found: 1 requests

Showing 1 to 1 of 1 rows

## Field Data

Each **Request** is stored in the database on the Request table. The field data can be in the Data column and is represented in XML.

- **Request** – Request root.
- **Field** – Each field is represented by a Field element. The value of the field is HTML-Encoded and has the following attributes.
  - **Id** – Unique identifier of the field.
  - **Name** – HTML-Encoded name of the field.
  - **Type** – The field type. Valid values: TextBox, TextArea, Hidden, CheckBox, RadioButton, ListSingleSelection, ListMultipleSelection, Date, Time, TableSingleSelection, TableMultipleSelection

```
<Request>
  <Field Id="1" Name="Your Text Box Field" Type="TextBox">Your Value</Field>
</Request>
```

## Dashboard

The Dashboard allows you to view the current state of requests for your Services and Offerings.

- **Total Request** – Displays the number of total requests.
- **Total Cost Savings** – Displays the cost saved of complete or closed requests.
- **Total Time Saved** – Displays the time saved of complete or closed requests.
- **Requests by State** – Displays a pie chart of requests by state.
- **Top Services by Requests** – Displays a bar chart of top services by requests.
- **Top Offerings by Request** – Displays a bar chart of top offerings by requests.
- **Top Offerings by Cost Saved** – Displays a bar chart of the cost saved for complete or closed requests.
- **Top Offerings by Time Saved** – Displays a bar chart of the time saved for complete or closed requests.
- **Top Users by Requests** – Displays a bar chart of the top users by requests.

**Note:** All panels include deleted requests.

## Dashboard

Total Requests

2

Total Cost Savings

\$110.00

Total Time Saved

2 hours, 0 minutes

Period

24 Hours

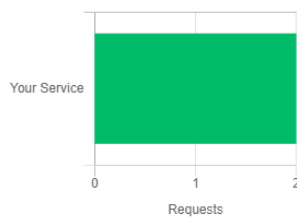
Last Updated

5/12/2021, 12:08:30 PM (Every 3 min.)

Requests By State



Top Services by Requests



Top Offerings by Requests



Top Offerings by Cost Saved



Top Offerings by Time Saved



Top Users by Requests



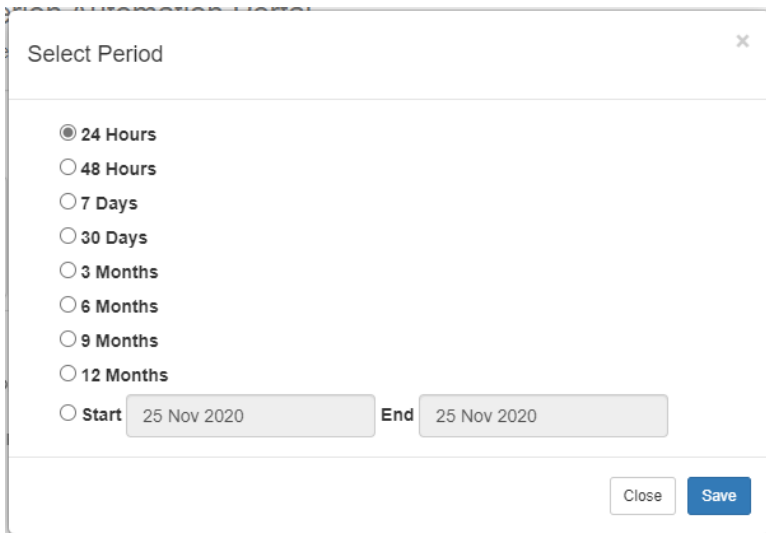
## Period

The time period of the dashboard can be changed by clicking the current period in the Period panel.

Period

24 Hours

Select a new time period from the available list. When a fixed period is selected the dashboard shows all past requests in that selected time windows. When using the custom start and end period all requests from start at midnight to end at midnight are displayed.



Select Period

☒ 24 Hours  
☐ 48 Hours  
☐ 7 Days  
☐ 30 Days  
☐ 3 Months  
☐ 6 Months  
☐ 9 Months  
☐ 12 Months  
☐ Start: 25 Nov 2020 End: 25 Nov 2020

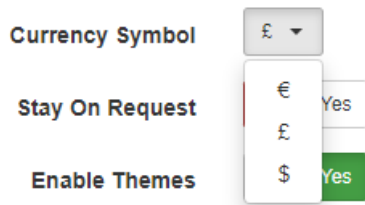
Close Save

## Last Updated

The update interval can be configured in the **Admin -> Settings**.

## Currency Symbol

The currency symbol can be changed in the Admin -> Settings panel.



Currency Symbol: £

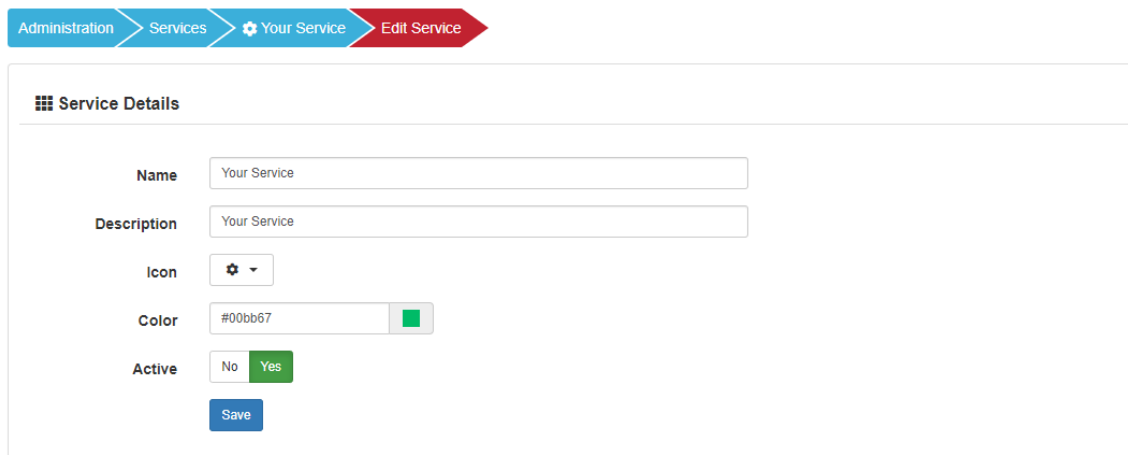
Stay On Request: € Yes

Enable Themes: £ Yes

\$ Yes

## Chart Color

The color of Service and Offering chart items can be changed in the Service Details and Offering Details.



Administration > Services > Your Service > Edit Service

**Service Details**

Name: Your Service

Description: Your Service

Icon: [Icon]

Color: #00bb67 [Color Picker]

Active: No Yes

Save

[Administration](#) > [Services](#) > [Your Service](#) > [Your Offering](#) > [Edit Offering](#)

### Offering Details

Name	<input type="text" value="Your Offering"/>
Description	<input type="text" value="Your Offering"/>
User Groups	<input type="text" value="Operator"/> +
Team Groups	<input type="text" value="Team"/> +
Requires External Ref. ID	<input checked="" type="radio"/> No <input type="radio"/> Yes
Requires Group Approval	<input checked="" type="radio"/> No <input type="radio"/> Yes
Cost Saved	<input type="text" value="\$ 55.00"/>
Time Saved	<input type="text" value="60"/> <input type="text" value="minutes"/>
Icon	<input type="text" value="⚙️"/>
Color	<input type="text" value="#f4903e"/> <input type="color" value="#f4903e"/>
Active	<input type="radio"/> No <input checked="" type="radio"/> Yes
Tag	<input type="text"/>
External Url	<input type="text"/>
Service	<input type="text" value="⚙️ Your Service"/>
Folder	<input type="text" value="Your Service Root"/>
<input type="button" value="Save"/>	

## Cost and Time Saved

A cost and time saved can be defined when creating an offering or updated at a later time.



Requires External Ref. ID

No Yes

Requires Group Approval

No Yes

Cost Saved

£ 0

Time Saved

0 minutes

Icon



Color

#00aabb

Active

No Yes

The cost and time saved panels in the dashboard show the savings for requests that have a state **Completed** or **Closed**.

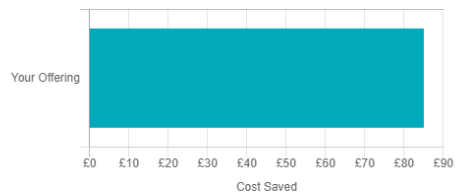
Total Cost Savings

£85.00

Total Time Saved

0 hours, 30 minutes

Top Offerings by Cost Saved



Top Offerings by Time Saved



## Customizing

### CSS

The style of the Automation Portal can be modified using the Site.css file located in the Content folder. Before modifying the Site.css you should make a backup. You can contact [support@kelverion.com](mailto:support@kelverion.com) for the original Site.css for your installed version of the portal.

## Automation Portal Integration Methodology

### Introduction

The Automation Portal was designed with integration into Automation in mind from its initial conception and throughout its development, whether your automation platform of choice is System Center Orchestrator or Azure Automation.

The interface from the Automation Portal to Orchestrator and Azure Automation is via the Portal application database.

To make interfacing with the portal easier we include our SQL Server Integration Module with a Runbook Studio product for Azure Automation. For Orchestrator we include the integrations packs for SQL Server, Data Manipulation and Runbook Management Integration Packs within the bundle when you purchase the Automation Portal.

Using these components together it becomes straightforward to build an easy to maintain solution that provides a rich set of capabilities to your users.

### Overview

From a high-level perspective, the process of integrating the Automation Portal and your automation platform looks like the diagram below.

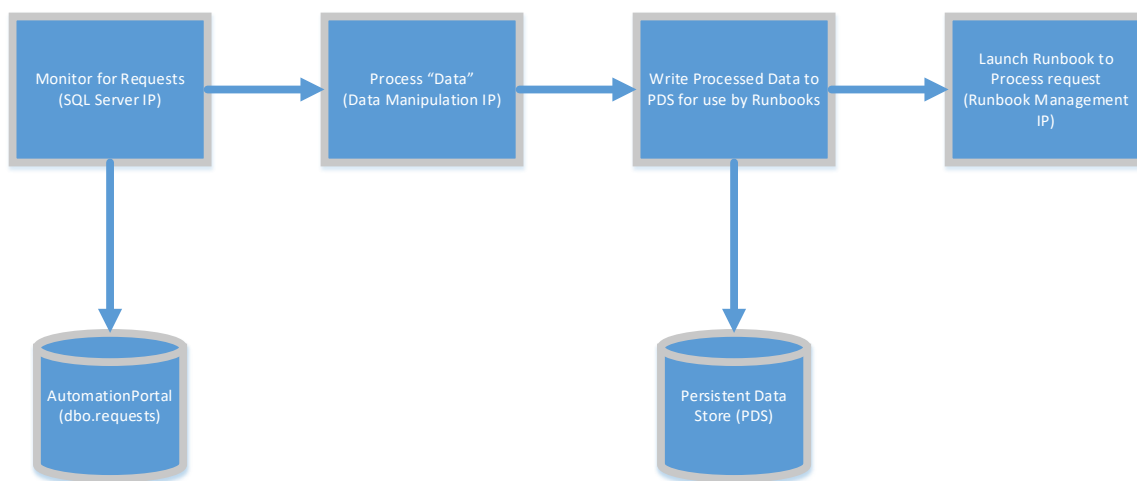


Figure 2 - Request Processing

We start by monitoring the Automation Portal database for requests that we can process. Each request contains the requestors input in XML, this allows the portal to be flexible and store all the diverse types of requests in a single table. When we have detected an appropriate request, we extract the data that is stored generically (in XML) and convert it into information that is consumable by our specific process. This data should then be logged into the Automation Persistent Data Store (PDS) finally we start the specific runbook to process the request.

## Generic Dispatcher ‘vs’ runbook specific dispatchers

When designing your runbook that integrates with the Automation Portal the first question is.

Do you need a generic dispatcher runbook that will pick up all the requests from the automation portal and launch the appropriate runbook, or will each runbook that processes data from the portal be monitoring the portal separately?

There are some tradeoffs with each approach. The table below lists the advantages and disadvantages.

Single monitoring dispatcher		Separate Monitors	
Advantages	Disadvantages	Advantages	Disadvantages
<b>Efficient</b>	More complex	Simplicity	Increased load
<b>Works well with existing runbooks</b>	Requires a mapping between service / offering and target runbook		May require modifying existing runbooks

Table 1 -Dispatcher Pros and Cons

The Kelverion best practice is to use a Single monitoring dispatcher as this is a more efficient and flexible approach the Portal Integration.

## Requests table

The Automation portal database contains a table called “Requests” which holds all the requests that have been created by users. The specific request data is held in the “data” column of the table in XML. The rest of the columns in the table contain the data about the request itself.

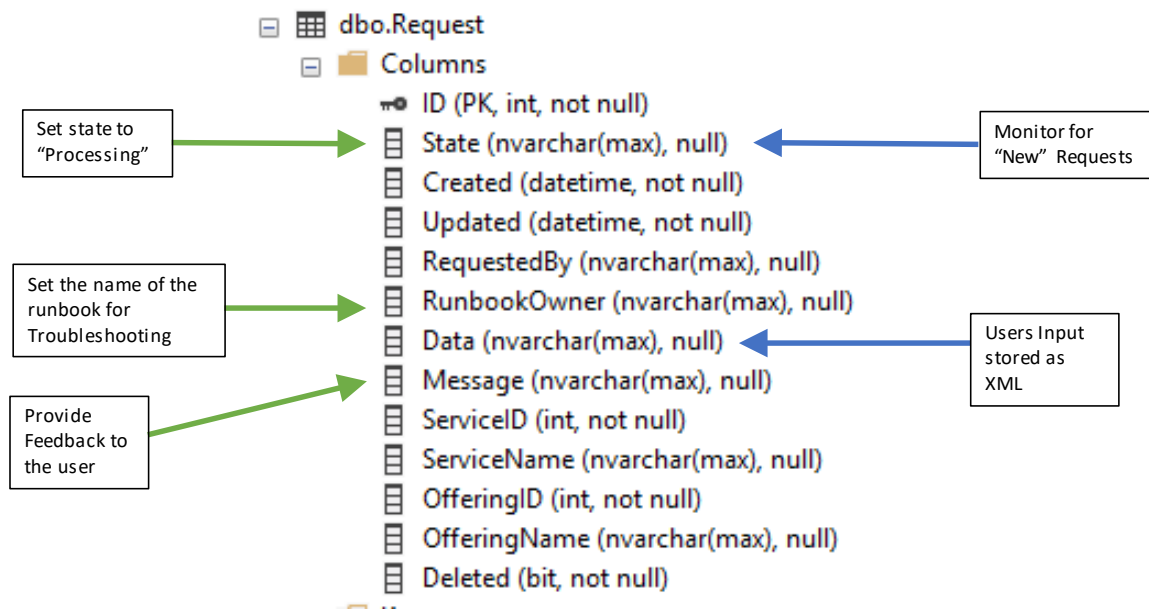


Table 2 - dbo.requests Table

The image above shows the columns that we interact with from our dispatcher runbook.

## Example Deployment using Orchestrator.

The approach to integration is broadly the same for both Azure Automation and Orchestrator so an Orchestrator example has been used to explain the process.

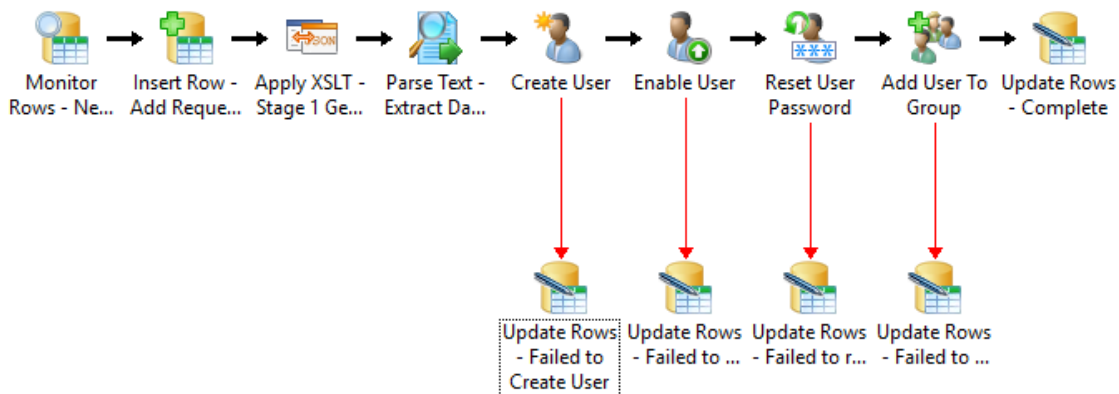
## A simple request monitoring runbook

We will look at a real example of integrating an Orchestrator runbook with the Automation Portal to allow users to request the creation of a new user account. In the real world, a new user onboarding process will be far more complex, but this example helps us to understand the interaction between the components of a typical solution.

Our runbook does the following.

- Identifying a new request
- Extracts the request data.
- Performs some processing.
- Provides feedback to the portal on the request state.

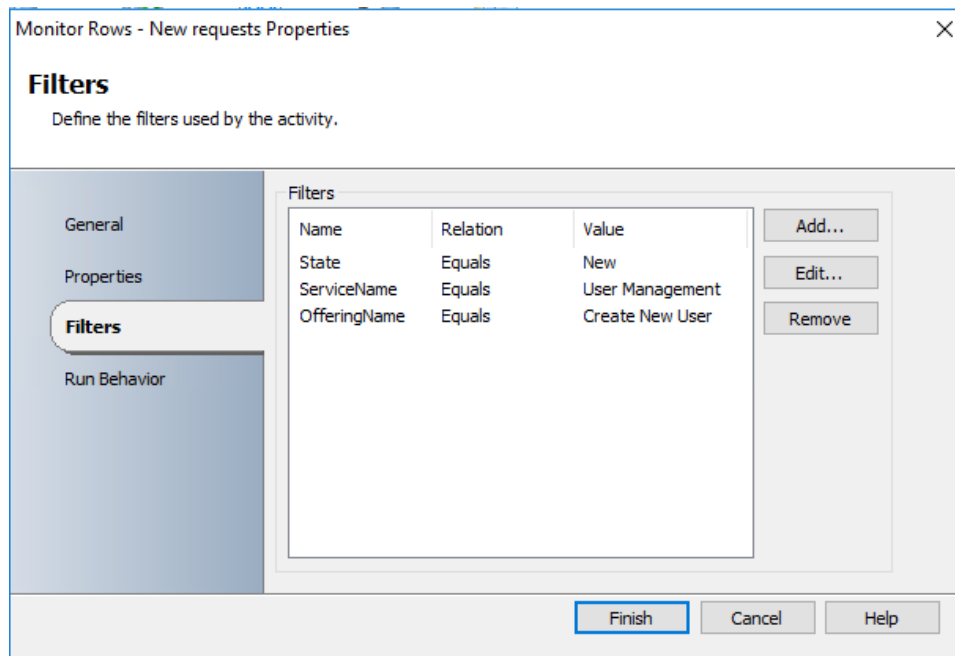
The runbook is show below:



## Identifying a new request

Using the Monitor rows activity from the SQL Server integration pack we Monitor for rows in the request table where the following is true by setting the object filters.

- The state is **New**.
- The service name is User **Management**.
- The Offering Name is **Create New User**.



When a matching record is found, the activity will update the following.

- State to **In Progress** (to ensure that the next time the monitor runs the row is not picked up again)
- The **RunbookOwner** to the name of the runbook (to help troubleshoot any failures)
- The **Message** to provide feedback to the user.

**Monitor Rows - New requests Properties**

**Properties**  
Define the properties used by the activity.

**General**  
**Properties**  
Filters  
Run Behavior

**Configuration**  
Name: AutomationPortal

**Properties**

Table/View Name	dbo.Request
Monitor Interval	15
State	In Progress
RunbookOwner	Monitor Automation Portal
Message	Automated Processing Started

Optional Properties...

Finish Cancel Help

After the request has been picked up the information is logged into the PDS for troubleshooting in this case, but in a more complex runbook the PDS is likely to be integral to the successful processing of the request. You should also consider inserting a record into the request history table to give the end user feedback about the status of their request.

## Extracting the request data.

The request data is held in XML, this allows the portal to be flexible in the way that requests are constructed and use a single table for storing the request data (without resorting to dynamic columns or other techniques that make processing the data in more challenging.) For the following offering:

**ApplicationTitle**  
Search

Vendor	Package_Title	Version	License_Cost
Microsoft	Visio	2010	£78

**User**  
Search

UserName	LineManager	CostCode
rcalley	Greg Charman	001122
swHITE	Greg Charman	001122

**Cost Code**  
Enter your cost code

**Justification**  
Please provide a valid business reason for the request

**Installation Date**  
Please provide the date the application is required from

**Removal Date**  
You must set this to a minimum of 2 weeks from the installation date

Submit

The XML for the request looks this:

```
<Request>
  <Field Id="4329"
Name="ApplicationTitle">&lt;table&gt;&lt;row&gt;&lt;Vendor&gt;Microsoft&lt;/Vendor&gt;&lt;
Package_Title&gt;Visio&lt;/Package_Title&gt;&lt;Version&gt;2010&lt;/Version&gt;&lt;License
_Cost&gt;£78&lt;/License_Cost&gt;&lt;state&gt;True&lt;/state&gt;&lt;/row&gt;&lt;/table&gt;
</Field>
  <Field Id="4330"
Name="User">&lt;table&gt;&lt;row&gt;&lt;UserName&gt;swhite&lt;/UserName&gt;&lt;LineManager
&gt;Mike
Smith&lt;/LineManager&gt;&lt;CostCode&gt;001122&lt;/CostCode&gt;&lt;state&gt;True&lt;/stat
e&gt;&lt;/row&gt;&lt;/table&gt;</Field>
  <Field Id="4331" Name="Cost Code">112233</Field>
  <Field Id="4332" Name="Justification">Testing the service</Field>
  <Field Id="4333" Name="Installation Date">2017-10-20</Field>
  <Field Id="4334" Name="Removal Date">2017-11-07</Field>
</Request>
```

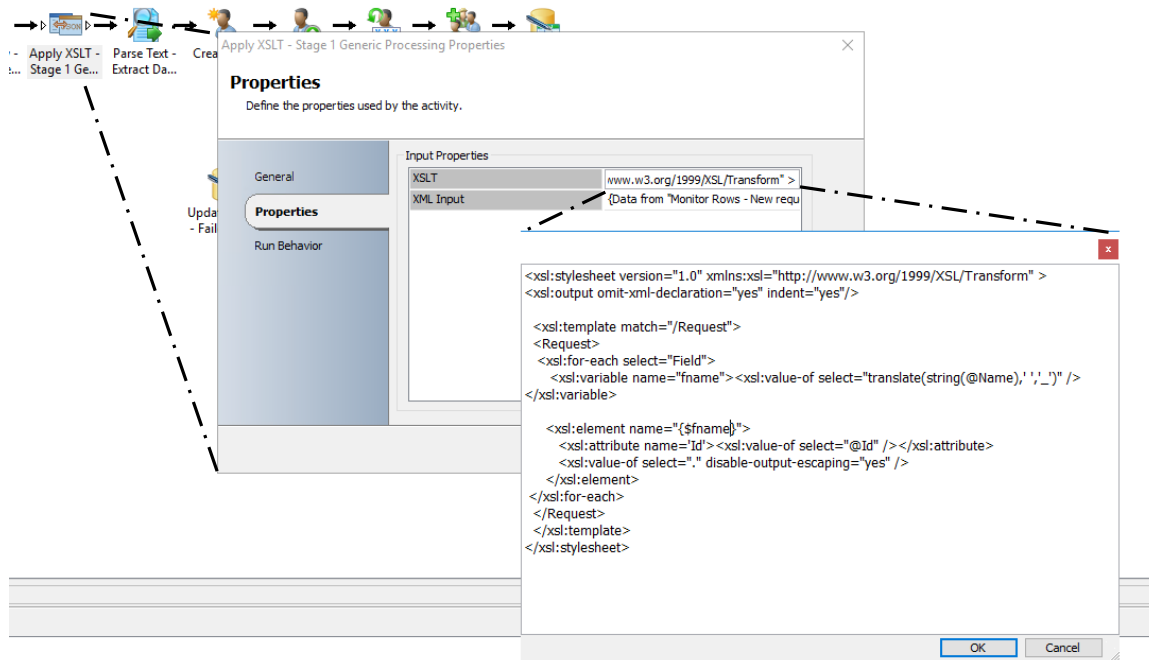
You may notice that the tables in the automation portal led to request data recorded as encoded XML (the normal XML < and > characters are replaced by &lt; and &gt; along with some other special characters)

The easiest way to convert these characters back to something we can process more easily in Orchestrator is by using the “Apply XSLT” activity from the data manipulation integration pack.

The following XSLT stylesheet will take care of the transformation process for you in both Azure Automation and Orchestrator.

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >
<xsl:output omit-xml-declaration="yes" indent="yes"/>
  <xsl:template match="/Request">
    <Request>
      <xsl:for-each select="Field">
        <xsl:variable name="fname"><xsl:value-of select="translate(string(@Name),' ','_')"/>
      </xsl:variable>
      <xsl:element name="{ $fname }">
        <xsl:attribute name="Id"><xsl:value-of select="@Id" /></xsl:attribute>
        <xsl:value-of select="." disable-output-escaping="yes" />
      </xsl:element>
    </xsl:for-each>
  </Request>
</xsl:template>
</xsl:stylesheet>
```

In your Orchestrator runbook, the Apply XSLT Activity will look like this:



The result of the transformation is the data that will now look like this:



```

<Request>
  <ApplicationTitle Id="4329">
    <table>
      <row>
        <Vendor>Microsoft</Vendor>
        <Package_Title>Visio</Package_Title>
        <Version>2010</Version>
        <License_Cost>£78</License_Cost>
        <state>True</state>
      </row>
    </table>
  </ApplicationTitle>
  <User Id="4330">
    <table>
      <row>
        <UserName>swhite</UserName>
        <LineManager>Mike Smith</LineManager>
        <CostCode>001122</CostCode>
        <state>True</state>
      </row>
    </table>
  </User>
  <Cost_Code Id="4331">112233</Cost_Code>
  <Justification Id="4332">Testing the service</Justification>
  <Installation_Date Id="4333">2017-10-20</Installation_Date>
  <Removal_Date Id="4334">2017-11-07</Removal_Date>
</Request>

```

In Orchestrator we can now use the Parse Text Activity from the Data Manipulation integration pack to publish named data onto the Databus.

The Parse Text Activity use a configuration file on your runbook server to specify exactly how to process the incoming data. It is a powerful activity that allows flexible extraction of data from XML using XPath, as well as the ability to use regular expressions and other techniques.

From the XML above if we wanted to publish the Application Vendor, the Package Title and the installation and removal dates onto the databus, then the following table shows the XPath expressions we would use.

Field	XPath
Vendor	/Request/ApplicationTitle/table/row/Vendor
Package_Title	/Request/ApplicationTitle/table/row/Package_Title
Installation_Date	/Request/Installation_Date
Removal_Date	/Request/Removal_Date

In the Data Manipulation configuration file that would look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<ka:DataManipulation xmlns:ka="http://www.kelverion.com">
  <ka:ParseText name="PackageRequest">
    <ka:Output name="Vendor"
      correlated="yes">/Request/ApplicationTitle/table/row/Vendor</ka:Output>
    <ka:Output name="Package_Title"
      correlated="yes">/Request/ApplicationTitle/table/row/Package_Title</ka:Output>
    <ka:Output name="Version"
      correlated="yes">/Request/Installation_Date</ka:Output>
    <ka:Output name="Device"
      correlated="yes">/Request/Removal_Date</ka:Output>
  </ka:ParseText>
</ka:DataManipulation>
```

Now we have the request data published onto the databus in an easy to consume form the runbook can complete its processing.

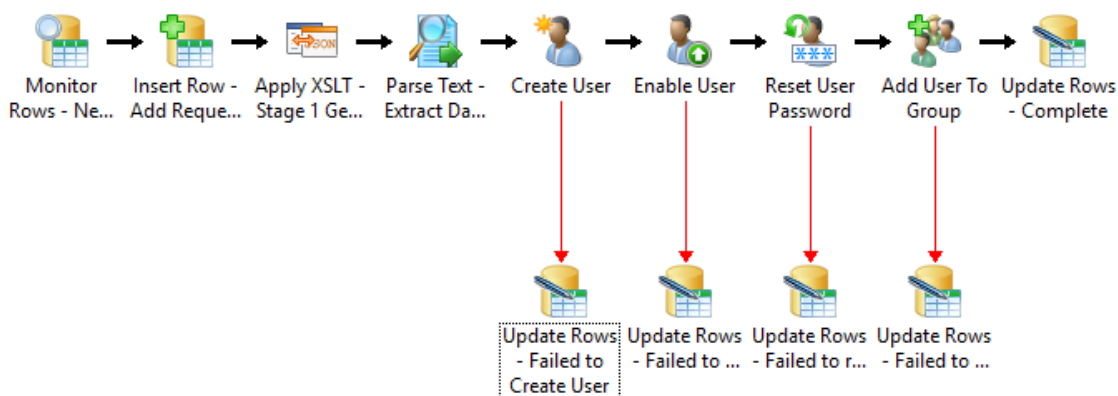
## Updating the Request status

Once the real work of the runbook has completed, it is important to update the status of the request with the final state, and to set an appropriate message for the user.

If the request has been processed successfully you should set the request state to **Completed**, and if there has been a problem with the processing of the request you should set the state of the request to **Failed**.

You can see those state changes being made using the SQL Server integration pack in the following picture.

Whenever you update the status of the request, or the message you should also insert a record into the request history table so that end users can see the actions that have completed as well as the overall status.



## Best Practice - Single Dispatcher runbook for multiple services and offerings

The best practice is to build a single dispatcher that will monitor requests across a spectrum of services and offerings. The dispatcher will contain some logic to identify the correct runbook to execute and start that runbook via the Runbook Management Integration Pack.

To follow this approach, it is important that each of the runbooks that is triggered has the same set of inputs. If the runbooks each have different inputs, then the dispatcher runbook will become complex and difficult to manage due to the way that the Orchestrator Web server works.

The easiest way to achieve consistent inputs for all your runbooks is using the PDS. Rather than passing the actual request data to the **worker runbooks** the dispatcher writes the request data to the PDS and then passes the ID of the PDS record to the runbook. This way all runbooks accept a single parameter, the PDS\_ID of the input data.

The following runbook is a general-purpose dispatcher using a database table to lookup the runbook to be executed, and the Runbook Management IP to launch the runbook.

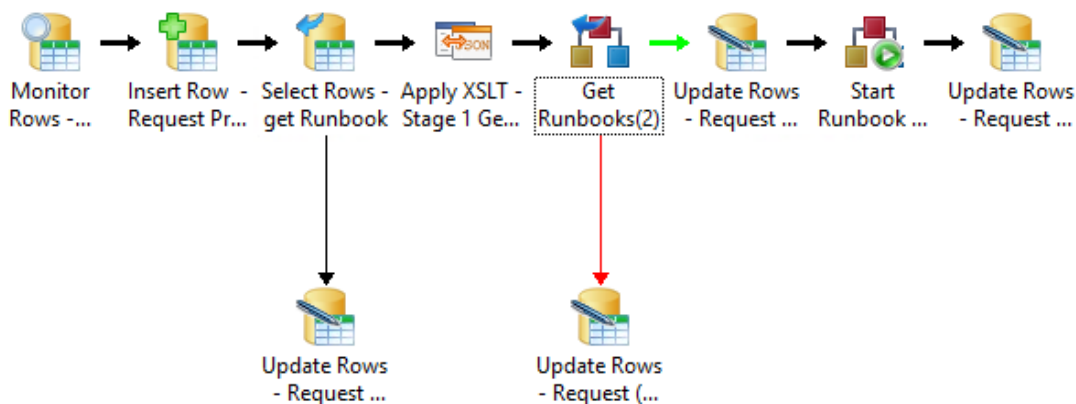


Table that maps between Service and Offering names, and the runbooks to execute looks like:

```
SELECT [ServiceName]
, [OfferingName]
, [RunbookPath]
, [Version]
FROM [DCC_PDS].[PortalData].[OfferingConversionMap]
```

	ServiceName	OfferingName	RunbookPath	Version
1	Active Directory	User\Create	\\Standard\Library\AD\User\AD-User-Create	1
2	Active Directory	User\Modify	\\Standard\Library\AD\User\AD-User-Modify	1
3	Active Directory	User\Remove	\\Standard\Library\AD\User\AD-User-Remove	1

The “Select Rows – Get runbook” has the following filters configured:

Select Rows - get Runbook Properties

### Filters

Define the filters used by the activity.

General

Properties

**Filters**

Run Behavior

Name	Rela...	Value
OfferingName	Equals	{OfferingName from "Monitor ..
ServiceName	Equals	{ServiceName from "Monitor R.
_state	Doe...	Disabled

Add... Edit... Remove

Finish Cancel Help

And the **Get Runbooks** activity is used to retrieve the runbook that matches the path from the database and publish the runbook ID onto the Databus to be consumed in the **Start Runbook** activity.

Get Runbooks(2) Properties

### Filters

Define the filters used by the activity.

General

**Filters**

Run Behavior

Configuration

Name: RM

Name	Relation	Value
Path	Equals	{RunbookPath fro...

Add... Edit...

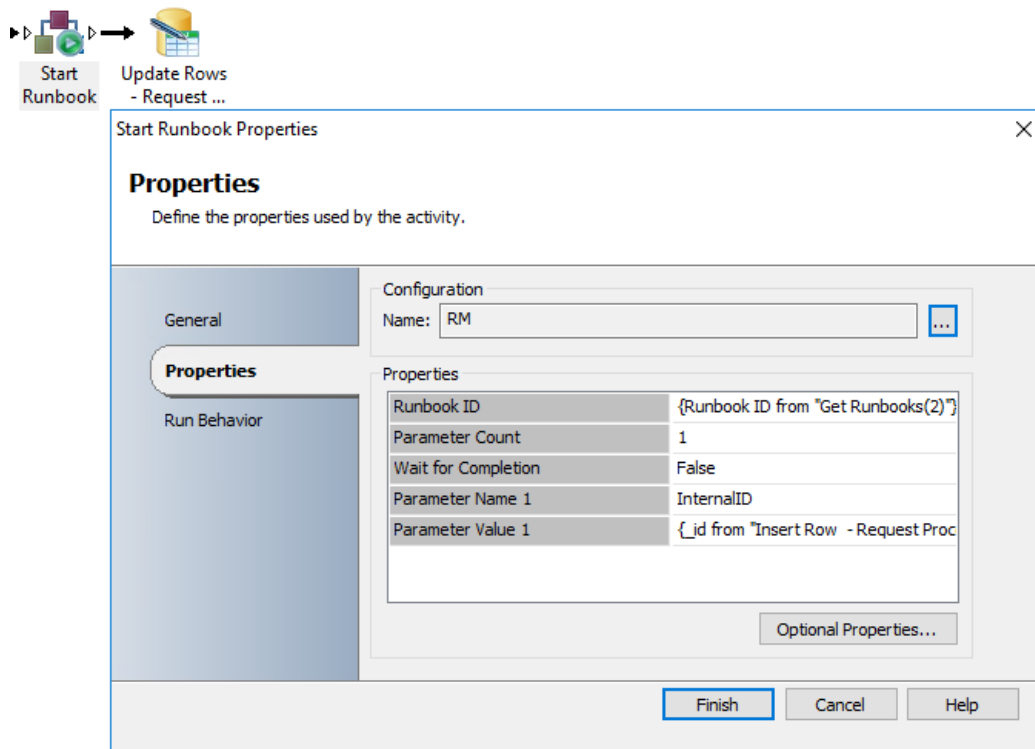
Filter Settings

Name: Path

Relation: Equals

Value: {RunbookPath from "Select Rows - get

OK Cancel



The other activities in the dispatcher follow the same pattern as the simple monitor runbook. It is important to ensure that whenever the **worker runbook completes** that request in the automation portal is still updated to ensure that the user is informed of the status of their request.

## Support and Guidance

---

If you require any support or guidance with deploying the portal and integrating it with your automation tool, our professional services team are ready to help.

Please contact your local sales representative who can organize a services support engagement.

## Sending Feedback

---

If you encounter a problem while working with the Automation Portal or have an idea for making the Automation Portal even better, we would like to hear from you.

You can send us an e-mail at [support@kelverion.com](mailto:support@kelverion.com)

We look forward to hearing from you.