

Control Room APIs

March 29, 2023



Legal Notices

© 2023 Automation Anywhere, Inc. All Rights Reserved.

See the list of Automation Anywhere trademarks at <https://www.automationanywhere.com/trademark>.

All other customer or partner trademarks or registered trademarks are owned by those companies.

The information contained in this documentation is proprietary and confidential. Your use of this information and Automation Anywhere Software products is subject to the terms and conditions of the applicable End-User License Agreement and/or Nondisclosure Agreement and the proprietary and restricted rights notices included therein.

You may print, copy, and use the information contained in this documentation for the internal needs of your user base only. Unless otherwise agreed to by Automation Anywhere and you in writing, you may not otherwise distribute this documentation or the information contained here outside of your organization without obtaining Automation Anywhere's prior written consent for each such distribution.

Examples and graphics are provided only as reference information and might not match your site.

Content

API references.....	6
Configure the Control Room.....	8
Authentication API.....	9
Authenticate (username and password).....	10
Authenticate (username and apiKey).....	14
Validate token.....	19
Refresh token.....	20
Expire token.....	26
User management APIs.....	27
Create role.....	30
List roles.....	38
Retrieve role.....	42
Update role.....	48
Delete role.....	58
Create user.....	58
Search for users API.....	65
Retrieve a specific user details API.....	69
Update an existing user details API.....	73
Delete an existing user API.....	76
Roles and permissions.....	78
Audit API.....	105
Example of createdOndate and userName filters in Audit API.....	109
Device API.....	112
List available unattended Bot Runners API.....	112
Assign default device API.....	115
Trigger API.....	116
Create an event trigger.....	117
Delete an event trigger.....	119
Credential Vault APIs.....	120
Set device login credentials API.....	122
Configure a locker using API.....	123
Configure a credential with attribute values using API.....	126
Assign credential to locker API.....	132
List credentials using API.....	132
Update attribute values.....	138
Get Masked credentials.....	140
Bot Execution Orchestrator API.....	144
Request device details.....	144
Activity list.....	149

Bot Scheduler APIs.	158
Schedule bot to run API.	159
List automation schedules API.	164
Repository Management APIs.	169
List files API.	172
List files and folders in a specific folder API.	177
List files and folders by workspace API.	180
Get Immediate Parents API.	183
Delete file/folder API.	186
Bot Insight API.	187
Get task log data.	188
Delete task log data.	191
Get bot run data.	193
Get Bot Insight audit trail data.	198
Get bot variables data.	200
Bot Lifecycle Management API.	202
Export files using API.	203
Import files using API.	205
Device pool API.	207
List device pools API.	207
Retrieve details of device pool by ID.	209
Create device pool API.	211
License API.	214
Retrieve Control Room license details API.	215
List Control Room licenses.	216
Deploy bots using API.	221
Bot deployment - V3.	222
Workload Management API.	231
Create Work Item model API.	232
Create queues API.	235
Add queue owner or member API.	237
Add queue participants API.	239
Add queue consumer API.	241
Add Work Items to the queue API.	243
Run bot with queue API.	247
Workload Management list APIs.	250
Migration APIs.	262
Start migration API.	263
List migration results API.	268
Bot migration results by id API.	271
Migration action mapping results API.	273
Enterprise 10 Migration APIs.	279

Filtering, pagination, and sorting.	300
API response codes.	307
Comparing Automation 360 and Enterprise 11 APIs.	308
Bot Agent API: Auto registration.	316

Control Room APIs

Control Room APIs

The Automation Anywhere Control Room provides APIs that allow you to customize how you (and your bots) interact with Automation Anywhere.

You use Control Room APIs to:

- Manage bot deployments
- Create and manage credentials in the Credential Vault
- Create and manage user accounts and roles
- Create and manage queues

Getting started with Control Room APIs

The Control Room APIs are built on HTTP and uses REST APIs with GET, POST, PUT, and DELETE methods. All requests must include an X-Authorization header with the JSON authentication token, or an **Authorization** header with a Bearer token for requests to the Control Room. The following sections provide details about the Control Room APIs available endpoints, methods, resources, authentication protocols, parameters, and headers, as well as examples of common requests and responses.

Note: The Bearer token is supported as of the Automation 360 v.27 release and later. It is unsupported and ignored in any previous releases. You must obtain the Bearer token from our OAuth services. To authorize your access, use either **X-Authorization** (using Authentication API) or **Authorization** (using OAuth). You cannot use both in the same API.

The Control Room supports several API clients. To explore the Control Room APIs, download and install [Postman](https://www.postman.com/collections/bb5cd3541d29e192cd43), then click <https://www.postman.com/collections/bb5cd3541d29e192cd43> to download the Control Room APIs collection. The following sections provide useful information about how to write a Control RoomAPI request:

- Some APIs require you to filter, paginate, and sort the results to get the desired output. For more information on filtering, pagination, and sorting.
- API response codes in the responses are described in the topic.

To learn how to test Control Room APIs in Swagger and Postman, watch the following video:
Learn about APIs

- **Configure the Control Room**

Before you start creating workflows for bot deployment and workload management using Control Room APIs, ensure you complete specific prerequisites and steps in the Control Room.

- **Authentication API**
Use the Authentication API to generate, refresh, and manage the JSON Web Tokens (JWTs) that are required for authentication and authorization in order to use the Control Room APIs.
- **User management APIs**
Use User Management APIs to create, search, update, or delete roles and users in your Control Room.
- **Audit API**
Use the Audit API to request audit data for a given input combination of date filter, sorting mechanism, and pagination.
- **Device API**
Identify all available users with unattended Bot Runner licenses, or filter for users by name.
- **Trigger API**
Map triggers to users or roles for an attended Bot Runner user by using the Trigger API. With the Trigger API, you can also create and delete event triggers.
- **Credential Vault APIs**
As an Control Room user with **Manage my credentials and lockers** feature permissions, you have the option to use the Credential Vault API to manage your attributes, credentials, and lockers in the Control Room.
- **Bot Execution Orchestrator API**
As a Control Room administrator or a user with **View and Manage Scheduled Activity** permission, you can monitor the bot progress using a set of Control Room APIs.
- **Bot Scheduler APIs**
Use the Bot Scheduler APIs to create, update, delete, and return details on scheduled automations.
- **Repository Management APIs**
Use the Repository Management APIs to return information on or to delete the objects (bots, folders, and files) that you have permissions to access in the Control Room.
- **Bot Insight API**
Users with the `AAE_Bot Insight Admin` or `AAE_Admin` role and the `Bot Insight` license can access the Bot Insight API to retrieve business and operations data.
- **Bot Lifecycle Management API**
Use the Bot Lifecycle Management API to export and import bots with dependent files and command packages for comprehensive automation lifecycle management. Users can export bots from public workspace and import to a private workspace in another Control Room and check into a public workspace.
- **Device pool API**
Identify all available device pools or filter device pools by name. Retrieve detailed device pool information for a device by searching for its unique numeric identifier (ID).
- **License API**
The License API contains endpoints to retrieve Control Room license details (such as expiration date and license mode) and manually sync the Control Room with the license server after license reallocation or renewal.
- **Deploy bots using API**
Use a combination of endpoints to deploy bots from the public workspace to Bot Runner devices.

- **Workload Management API**

Use the Workload Management API to programmatically manage and create Work Item models, queues, Work Items, and automations in your Control Room.

- **Migration APIs**

Use migration APIs to migrate MetaBots and TaskBots that were created in Enterprise client versions Enterprise 11 and Enterprise 10 to Automation 360. Use this page to review the migration prerequisites and access Enterprise 11 and Enterprise 10 Migration APIs.

- **Filtering, pagination, and sorting**

The Control Room API supports filtering, pagination, and sorting for endpoints that return arrays of resources.

- **API response codes**

Review the HTTP status codes of responses for Automation 360 APIs.

- **Comparing Automation 360 and Enterprise 11 APIs**

Compare Automation 360 and Enterprise 11 APIs to understand the contract changes when you migrate from Enterprise 11 to Automation 360.

- **Bot Agent API: Auto registration**

Automatically register and connect your device to a Control Room by using the Auto registration API.

Configure the Control Room

Before you start creating workflows for bot deployment and workload management using Control Room APIs, ensure you complete specific prerequisites and steps in the Control Room.

Prerequisites

Ensure you have following licenses and permissions:

- One or more Bot Creator and unattended Bot Runner licenses.
- Control Room admin credentials to view, create, and configure users, roles, and device pools.
- The **Create device pools** feature permission or the **AAE_Pool Admin** role must be assigned to you.

Procedure

Follow these steps to configure your Control Room with users, roles, and bots.

1. Log in to the Control Room as an admin.
2. Create a Control Room user.
Ensure that the user is assigned the **AAE_BASIC** role.
[Create a user](#)
3. Create a custom role to map the users you created to Bot Runners and Bot folders.
 - a. In the **Features** tab, ensure you select these permissions: **View my Scheduled Bots**, **Schedule my bots to run**, **Run my Bots**, **View my Bots**, and **Generate my API Key**.

- b. In the **Automation** tab, expand the **Bots** folder and select the root folder or subfolders you want to provide access for use. Ensure you select these permissions: **Run and schedule** and **View content**.
 - c. In the **Run as** tab, select one or more Bot Runners from the list of **Available bot runners**.
 - d. In the **Users** tab, select the new user created in Step 1 to assign the custom role to this new user.
4. Create a new device pool.
In the **Consumers** tab, ensure you select the custom role you created in Step 3.
[Create device pools](#)
5. Log in to the Control Room as a Bot Creator and create bots to automate routine business tasks.
[Build a Go be Great bot](#)
Ensure you check in the bots in the Bot folder or folders selected when creating the custom role (Step 3). [Check in a bot](#)

Parent topic: [Control Room APIs](#)


Authentication API

Use the Authentication API to generate, refresh, and manage the JSON Web Tokens (JWTs) that are required for authentication and authorization in order to use the Control Room APIs.

 **Note:** You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

Set up authentication or authorization for your application

1. Generate a token with one of the following endpoints:

 **Note:** `apiKey` or `password` is used to generate a valid token. Both (`apiKey` and `password`) cannot be used together in a request body.

- [Authenticate \(username and password\)](#)

If you are trying out the Control Room APIs in Swagger or another REST client, use this authentication method.

- [Authenticate \(username and apiKey\)](#)

Use this authentication method to generate the token without the need for the user's password, such as for organizations that use single sign-on (SSO).

⚠ **Note:** Tokens have a default timeout of 20 minutes. Do not use an expired token for your API requests. Using an expired token will invalidate the current valid token.

2. [Validate token](#)
3. [Refresh token](#)
4. [Expire token](#)

Authenticate (username and password)

Use this API to authenticate access to your Control Room with a valid `username` and `password`. A successfully completed response generates a JSON Web Token. By default, a token is valid for 20 minutes.

Request

```
POST http://{{ControlRoomURL}}/v1/authentication
```

Request body:


```
{
  "username": "jdoe",
  "password": "mypassword@123"
}
```

Request body to generate Multi-login token:

```
{
  "username": "jdoe",
  "password": "mypassword@123",
  "multipleLogin": true
}
```

⚠ **Note:** `apiKey` or `password` is used to generate a valid token. Both (`apiKey` and `password`) cannot be used together in a request body.

Request Parameters

Parameter	Type	Description
username	String	Enter your user name.
password	String	Enter your password.
multipleLogin	Boolean	<p>Allow or disallow multiple login. For more information on multi-login, see Multi-login user.</p> <ul style="list-style-type: none">• <i>true</i> - Allows multiple login• <i>false</i> - Disallows multiple login <div> Note: If this value is set to <i>true</i>, you will be allowed multiple API sessions.</div>

Response

```
{
  "token": "eyJhbGciOiJSUzUxMiJ9.eyJzdWIiOiIxIiwiaWY2xpZW50VHlwZSI6IldFQiIsImxpY2Vuc2VzIjpbXSwiYW5hbH10aWNzTG1jZW5zZXNqdXJjaGFzZWQiOnsiQW5hbH10aWNzQ2xpZW50Ijpb0cnVlLCJBbmFseXRpY3NBUEkiOnRydWV9LCJpYXQiOiJlNzMxMDE4NDZMsImV4cCI6MTUzMzEwOTA3MiwiaXNzIjoiaXQV0b21hdGlvbkFueXdoZXJlIiwibmFub1RpbWUiOiJmM2NTc1NjI0OTQ2MzE2MDAsImNzcmZUb2t1biI6ImNiZjgwZW5kZmU5YmUwYzViOGI2MDk3NmU0ZTI2MTNiIn0.rGYxbS5kKUTxtZhYtRSXpmIHwbf3IwLBIKDEA7odG5uGVAjD55Tv05bYdArx_3-tl1CBg_cDGbwj5FvaBt9u5xKu5W5j3Nur6x3PF62NeB3ZIdxiUPaFBU0Br84mPJMD4_EpwBfbsVOMH6ngiLtJYhIOtJa0kp4pAAM3mvkuOUELtH8lf3pQf-2Ose2fUAaebDkqiH13SUF1TONAjUQv6Ef_uY0wgq9SjZwKHg9SKUhX3S8PXAjNe_ih2QnN8nUE1SXGlkC04eoIvyWpFkM963XEjptc2uvwtVn42MdA4Nd1opD5yijEl9VM92Fe1sPb6_T5-oV-U1Iw0JHiX2-Ug",
  "user": {
    "id": 9,
    "email": "a@a.com",
    "username": "jdoe",
    "domain": null,
  }
}
```

```
"firstName":"j",
"lastName":"doe",
"version":9,
"principalId":9,
"deleted":false,
"roles":[
  {
    "name":"API_Key_Generation",
    "id":23,
    "version":0
  },
  {
    "name":"AAE_Basic",
    "id":2,
    "version":0
  },
  {
    "name":"Docrole1",
    "id":18,
    "version":0
  }
],
"sysAssignedRoles":[

],
"groupNames":[


],
"permissions":[
  ". . ."
],
"licenseFeatures":[
  "RUNTIME"
],
"emailVerified":true,
"passwordSet":true,
```

```

    "questionsSet":true,
    "enableAutoLogin":false,
    "disabled":false,
    "clientRegistered":false,
    "description":"","
    "createdBy":1,
    "createdOn":"2022-03-10T13:39:56-05:00",
    "updatedBy":1,
    "updatedOn":"2022-03-13T02:09:38-05:00",
    "publicKey":null,
    "appType":null,
    "routingName":null,
    "appUrl":null
  }
}

```

Response Parameters

Parameter	Type	Description
token	String	<p>Generated access token that acts as a session ID that your application will use for making requests. This token is equivalent to the user credentials and must be protected.</p> <div>  Note: If <i>multipleLogin</i> is set to <i>true</i>, you will be able to use this token for multiple API sessions. </div>
user	Object	<p>The user object returned with all the details of the user.</p> <ul style="list-style-type: none"> • id - Id of the user. • email - Email id of the user. • username - User name of the user. • domain - Domain logged in to. • firstname - First name of the user. • lastname - Last name of the user. • roles - Roles assigned to the user. • sysAssignedRoles - System assigned roles.

Parameter	Type	Description
		<ul style="list-style-type: none"> permissions - Permissions assigned to the user. licenseFeatures - License assigned to the user.

Insert the token in the request header of subsequent API requests.

Note: For an Control Room that is deployed on Cloud and has SAML authentication enabled, generate the web token with your `username` and `apikey`.

[Authenticate \(username and apiKey\)](#)

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Authenticate (username and apiKey)

Use this API to authenticate access to a Control Room with the `username` and `apikey`. A successful response generates a JSON Web Token. A generated token is valid for 20 minutes. You can generate the token without the need for the user's password, such as for organizations that use single sign-on (SSO).

To generate the token, you require the following:

- A custom role with the Generate API-Key permission
- Valid `username` and `API-key` to the Control Room. The `API-Key` is a 40-character string generated in the Control Room.

[Create and assign API key generation role](#)

Request

```
POST http://{{ControlRoomURL}}/v1/authentication
```

Request body:

```
{
  "username": "jdoe",
  "apiKey": "Vie;Z:IvtAhY0\\1RAD[SWl{NU7baRLYEeIYUJSKO"
}
```

Note: With `apiKey` you will be generating a multi session token. `multipleLogin` parameter is not applicable while you use `apiKey` to login.

Note: `apiKey` or `password` is used to generate a valid token. Both (`apiKey` and `password`) cannot be used together in a request body.

Request Parameters

Parameter	Type	Description
username	String	Enter your user name.
apikey	String	Enter your apikey. <div> <p>Note: With .24 or previous versions, you must replace any escape character <code>\</code> with <code>\\</code> in the API key. With .25 or later versions, the API key will no longer have any escape characters(<code>\</code>).</p> </div>

Response

```
{
  "token": "eyJhbGciOiJSUzUxMiJ9.eyJzdWIiOiIxIiwiaWY2xpZW50VHlwZSI6IldFQiIsImxpY2Vuc2VzIjpbXSwiYW5hbHl0aWNzTG1jZW5zZXNQdXJjaGFzZWQlOnsiQW5hbHl0aWNzQ2xpZW50Ijpb0cnVlLCJBbmFseXRpY3NBUEkiOnRydWV9LCJpYXQiOiJlNzIxMjM0NDc4NDc0ImV4cCI6MTUzMTZlOTA3MwYwIHNzIjoiQXV0b21hdGlvbkFueXdoZXJlIiwibmFub1RpbWUiOiJmMjNTc1NjI0OTQ2MzE2MDAsImNzcmZUb2t1b2I6ImNiZjgwZW5kZmU5YmUwYzViOGI2MDk3NmU0ZTI2MTNiIn0.rGYxbS5kKUTxtZhYtRSXpmIHwbf3IwLBIKDEA7odG5uGVAjD55Tv05bYdARx_3-tl1CBg_cDGbwj5FvaBt9u5xKu5W5j3Nu
```

```
r6x3PF62NeB3ZIdxiUPaFBU0Br84mPJMD4_EpwBfbsVOMH6ngiLtJYhIOtJa0kp4pAAm3mvkuOUEL
tH8lf3pQf-2Ose2fUAaebDkqiH13SUF1TONAjUQv6Ef_uY0wgq9SjZwKHg9SKUhX3S8PXAJne_ih2Q
nN8nUE1SXGlkc04eoIvyWpFkM963XEjptc2uvwtVn42MdA4NdlopD5yijEl9VM92Fe1sPb6_T5-oV-
U1Iw0JHiX2-Ug",
  "user": {
    "id": 9,
    "email": "a@a.com",
    "username": "jdoe",
    "domain": null,
    "firstName": "j",
    "lastName": "doe",
    "version": 9,
    "principalId": 9,
    "deleted": false,
    "roles": [
      {
        "name": "API_Key_Generation",
        "id": 23,
        "version": 0
      },
      {
        "name": "AAE_Basic",
        "id": 2,
        "version": 0
      },
      {
        "name": "Docrole1",
        "id": 18,
        "version": 0
      }
    ],
    "sysAssignedRoles": [

  ],
    "groupNames": [
```



```

    ],
    "permissions": [
        ". . ."
    ],
    "licenseFeatures": [
        "RUNTIME"
    ],
    "emailVerified": true,
    "passwordSet": true,
    "questionsSet": true,
    "enableAutoLogin": false,
    "disabled": false,
    "clientRegistered": false,
    "description": "",
    "createdBy": 1,
    "createdOn": "2022-03-10T13:39:56-05:00",
    "updatedBy": 1,
    "updatedOn": "2022-03-13T02:09:38-05:00",
    "publicKey": null,
    "appType": null,
    "routingName": null,
    "appUrl": null
  }
}

```

Response Parameters

Parameter	Type	Description
token	String	Generated access token that acts as a session ID that your application will use for making requests. This token is equivalent to the user credentials and must be protected.
user	Object	<p>The user object returned with all the details of the user.</p> <ul style="list-style-type: none"> id - Id of the user. email - Email id of the user.

Parameter	Type	Description
		<ul style="list-style-type: none"> username - User name of the user. domain - Domain logged in to. firstname - First name of the user. lastname - Last name of the user. roles - Roles assigned to the user. sysAssignedRoles - System assigned roles. permissions - Permissions assigned to the user. licenseFeatures - License assigned to the user.

Insert the token in the request header of subsequent API requests.

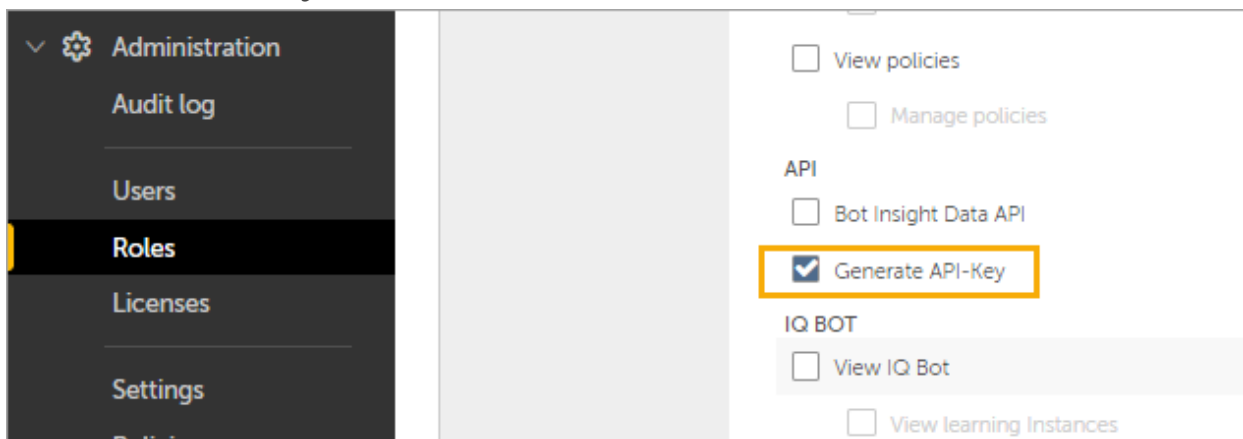
Create and assign API key generation role

As an Administrator, you can create a custom role to generate an API key and assign that custom role to users. By default, the `Generate API-Key` parameter is not enabled for any of the System-created roles.

To create and assign a custom role to generate an API key:

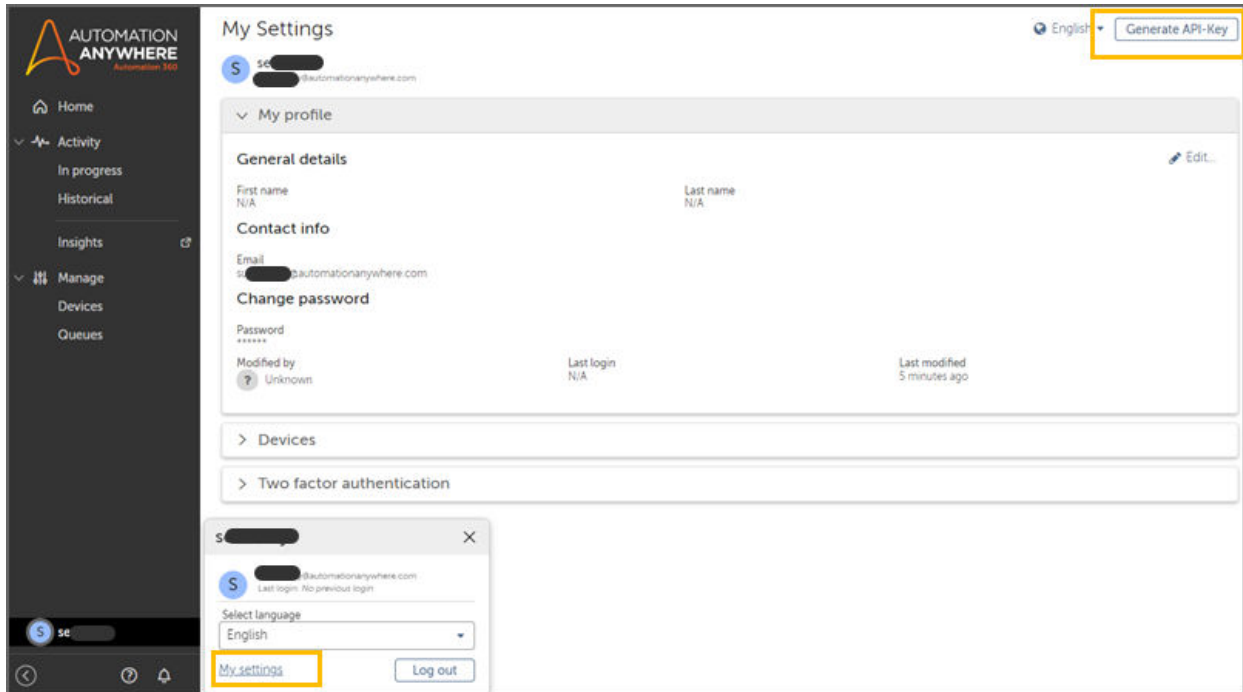
Procedure

1. Log in as an Administrator to the Automation 360 Control Room.
2. Navigate to: **Administration > Roles**.
3. Click **Create role**. For more information, see [Create a role](#).
4. Scroll down to the **API** section.
5. Select **Generate API-Key**.



6. Enter a unique name in the **Role name** field.
7. Click **Create role**.

8. Navigate to: **Administration > Users**, and assign the custom role you just created to a non-Administration user.
9. Log in as the user you assigned the **Generate API-Key** role to.
10. Navigate to the username profile at the bottom left of the page and select **username > My settings**.
11. Click **Generate API-Key** at the top right of the page.



12. Copy the generated **API-Key** to your clipboard.

Next steps

Use the **API-Key** to log in to a Control Room using SSO, or use the **API-Key** to log in as a user without a password.

Validate token

Verify if a JSON Web Token is valid.

Request

```
GET https://{ControlRoomURL}/v1/authentication/token?token=<token>
```

```
Header: X-Authorization <<authentication token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

If the token is not valid, you must generate a token using one of the authentication methods: [Authentication API](#)

Request Parameters

Parameter	Type	Required	Description
token	String	Yes	<p>Enter the token you want to validate.</p> <p>To generate a token, see Authenticate (username and password) or Authenticate (username and apiKey).</p>


Response

200 OK

```
{
  "valid": true
}
```

Response Parameters

Parameter	Type	Description
valid	Boolean	<p>Returns</p> <ol style="list-style-type: none">1. <code>true</code> , if the token is valid.2. <code>false</code> , if the token is invalid.

 **Note:** You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Refresh token

Generate a JSON Web Token using a valid existing token. This endpoint provides you with a new token without the need to collect and authenticate credentials every time a token expires. By default, a token is valid for 20 minutes.

Request

```
POST https://{ControlRoomURL}/v1/authentication/token
```

```
Header: X-Authorization <<authentication token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request body:

```
{
  "token": "eyJhbGciOiJSUzUxMiJ9.eyJzdWIiOiIxIiwia2xpZW50VHlwZSI6IldFQiIsImxwY2Vuc2VzIjpibXSwiYW5hbH10aWNzTG1jZW5zZXNqdXJjaGFzZWQiOnsiQW5hbH10aWNzQ2xpZW50Ijp0cnVlLCJBbmFseXRpY3NBUEkiOnRydWV9LCJpYXQiOiJlNzIxMzMDgwNjEsImV4cCI6MTU3MzEwOTIyMzswiaXNzIjoiQXV0b21hdGlvbkFueXdoZXJlIiwibmFub1RpbWUiOiJmM2NTc3NTA4OTY5NzUxMDAsImNzcmZUb2t1b2I6ImJiInJgZMGJhMDY5MwYyYjZiM2M3MDE4NGY0OGM0MwY1In0.f3kPRspfm0sei9DGHd9NoyLK-iCO-vs--8b_pLG9XSUR0186uvXFopB75eVAaG-1l_AZhR78UE6Voi7_UggzHkLRrEpQ-szR7cmFDpLxZ28xLnFJYhaIuMNdW9dWDVquBWTQSpYGNJd56D-tFFHBodwVdNamqWHxaQebq1zMyUyQV6Q-gKdgubpT5gwuXnp-BwScjHOYM3Fpj_nt0nEbJC5uWpJNtLQBpVzhsRwWlRKNOHQVbo6X7zkvKBoij8ewa5FWQwX7T-760BeqfssR6mmMU00zRaneUKAYAskz0B-X5PcyCkrVJju2XqItQ9XMGNP7h_MaUDotU_CJyguPZA"
```

Request Parameters

Parameter	Type	Required	Description
token	String	Yes	<p>Enter the token you want to refresh.</p> <p>If the token is not valid, you must generate a token using one of the authentication methods:</p>

200 OK

© 2023 Automation Anywhere. All rights reserved. 22

```
],
"sysAssignedRoles": [],
"groupNames": [],
"permissions": [
  {
    "id": 163,
    "action": "own",
    "resourceId": "1",
    "resourceType": "queue"
  },
  {
    "id": 141,
    "action": "cancelcheckout",
    "resourceId": null,
    "resourceType": "repositorymanager"
  },
  {
    "id": 165,
    "action": "own",
    "resourceId": "1",
    "resourceType": "pool"
  },
  {
    "id": 97,
    "action": "register",
    "resourceId": null,
    "resourceType": "devices"
  },
  {
    "id": 161,
    "action": "participate",
    "resourceId": "1",
    "resourceType": "queue"
  },
  {
    "id": 29,
```

```
        "action": "view",
        "resourceId": null,
        "resourceType": "repositorymanager"
    },
    {
        "id": 164,
        "action": "manage",
        "resourceId": "1",
        "resourceType": "pool"
    },
    {
        "id": 31,
        "action": "export",
        "resourceId": null,
        "resourceType": "repositorymanager"
    },
    {
        "id": 32,
        "action": "import",
        "resourceId": null,
        "resourceType": "repositorymanager"
    }
],
"licenseFeatures": [
    "DEVELOPMENT"
],
"emailVerified": true,
"passwordSet": true,
"questionsSet": true,
"enableAutoLogin": true,
"disabled": false,
"clientRegistered": false,
"description": "",
"createdBy": 1,
"createdOn": "2022-03-17T19:33:06Z",
"updatedBy": 1,
```




```

    "updatedOn": "2022-04-08T21:54:29Z",
    "publicKey": null,
    "appType": null,
    "routingName": null,
    "appUrl": null,
    "email": "jdoe@aa.com",
    "lastLoginTime": "2022-04-08T21:54:15Z",
    "deviceCredentialAttested": false,
    "multipleLoginAllowed": true
  },
  "tenantUuid": "282978c4-6386-c13a-92ac-5009e3cfd6b3",
  "mfaAuthResponse": null
}

```

Response Parameters

Parameter	Type	Description
token	String	<p>Generated access token that acts as a session ID that your application will use for making requests. This token is equivalent to the user credentials and must be protected.</p> <div>  Note: If <i>multipleLogin</i> is set to <i>true</i>, you will be able to use this token for multiple API sessions. </div>
user	Object	<p>The user object returned with all the details of the user.</p> <ul style="list-style-type: none"> • id - Id of the user. • email - Email id of the user. • username - User name of the user. • domain - Domain logged in to. • firstname - First name of the user. • lastname - Last name of the user. • roles - Roles assigned to the user. • sysAssignedRoles - System assigned roles. • permissions - Permissions assigned to the user.

Parameter	Type	Description
		<ul style="list-style-type: none"> licenseFeatures - License assigned to the user.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Expire token

Immediately log out the user and invalidate their JSON Web Token so that it cannot be used for authentication.

Request

```
POST https://{ControlRoomURL}/v1/authentication/logout
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request body:

```
{
  "token": "eyJhbGciOiJSUzUxMiJ9.eyJzdWIiOiIxIiwiaWF0Ij0iY2xpZW50VHlwZSI6IldlFQIiIsImxwY2Vuc2VzIjpbXSUiYW5hbH10aWNzTG1jZW5zZXNqdXJjaGFzZWQiOnsiQW5hbH10aWNzQ2xpZW50Ijp0cnVlLCJBbmFseXRpY3NBUEkiOnRydWV9LCJpYXQiOiJlNzIxMDE4MDgwNjEsImV4cCI6MTU3MzEwOTIyMzIwIiwiaXNzIjoiQXV0b21hdGlvbkFueXdoZXJlIiwibmFub1RpbWUiOiJmM2NTc3NTA0OTY5NzUxMDAsImNzcmZUb2t1biI6ImJinJgZMGJhMDY5MjYyYjZiM2M3MDE4NGY0OGM0MjY1In0.f3kPRspfm0sei9DGHd9NoyLK-iCO-vs--8b_pLG9XSUR0186uvXFopB75eVAaG-11_AZhR78UE6Voi7_UggzHkLRrEpQ-szR7cmFDpLxZ28xLnFJYhaIuMNdW9dWDVquBWTQSpYGNJd56D-tFFHBodwVdNamqWHxaQebq1zMyUyQV6Q-gKdgubpT5gwXnp-BwScjHOYM3Fpj_nt0nEbJC5uWpJNtLQBpVzhsRwwlRKNOHQVbo6X7zkvKBoij8ewa5FWQwX7T-760BeqfssR6mmMUo0zRaneUKAYAskz0B-X5PcyCkrVJju2XqItQ9XMGNP7h_M
```

```
aUDotU_CJyguPZA"
}
```

Request Parameters

Parameter	Type	Required	Description
token	String	Yes	<p>Enter the session's token that you want to logout.</p> <p>To generate a token, see Authenticate (username and password) or Authenticate (username and apiKey).</p>

Response

204 No Response

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

User management APIs

Use User Management APIs to create, search, update, or delete roles and users in your Control Room.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

User Management Roles

Users need the following permissions in order to create and manage users and roles.

- **View users**

You need an administrator permission to create and manage users and roles. It is recommended that non-administrator users be given limited permissions for creating and managing users. Learn how to [create a role with limited permissions](#) that can be assigned to users.

- **Create users**

Create new users in the Control Room.

- **Edit users**

Edit all users in the Control Room, including users created by other administrators.

- **Delete users**

Delete any user in the Control Room.

- **View roles**

Users with this permission can view roles to which they have access.

- **Manage roles**

Users can create, edit and delete roles to which they have access.

- **View licenses**

Users with these permissions are able to view and manage device licenses. Device licenses are required to enable users to perform specific tasks. For example, Bot Creators require a **DEVELOPMENT** device license in order to create bots.

- **Manage users device license**

Users with this permission can assign device licenses to other users.

Role APIs

Use Role APIs to List Roles, create a role, retrieve a specific role using an object ID, update a role, or delete a role.

List roles

Retrieves current roles based on search criteria, such as filtering, sorting, and pagination.

```
POST http://<your_control_room_url>/v1/usermanagement/roles/list
```

Create role

Creates a new role with a new role name.

```
POST http://<your_control_room_url>/v1/usermanagement/roles
```

Retrieve role

Retrieves a specific role based on a unique role ID.

```
GET http://<your_control_room_url>/v1/usermanagement/roles/{id}
```

Update role

Modifies an existing role name based on a unique role ID.

```
PUT http://<your_control_room_url>/v1/usermanagement/roles/{id}
```

Delete role

Deletes an existing role based on a unique role ID.

```
DELETE http://<your_control_room_url>/v1/usermanagement/roles/{id}
```

User APIs

Use User APIs to create a user, search for users, retrieve a user details based on a user ID, update a specific user details, or delete a user.

Create user

Creates a user with a new user name.

```
POST http://<your_control_room_url>/v1/usermanagement/users
```

Search for users API

Retrieves current users based on search criteria, such as filtering, sorting, and pagination.

```
POST http://<your_control_room_url>/v1/usermanagement/users/list
```

Retrieve a specific user details API

Retrieves user details based on a unique user ID.

```
GET http://<your_control_room_url>/v1/usermanagement/users/{uid}
```

Update an existing user details API

Modifies an existing user name based on a unique user ID.

```
PUT http://<your_control_room_url>/v1/usermanagement/users/{uid}
```

Delete an existing user API

Deletes an existing user based on a unique user ID.

```
DELETE http://<your_control_room_url>/v1/usermanagement/users/{uid}
```

Create role

Use Create role API to create a new role with permissions in the Control Room.

Request

```
POST https://{ControlRoomURL}}/
/v1/usermanagement/roles
```

Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request body:

```
{
  "name": "Trigger Manager",
  "description": "View and Manage the triggers",
  "permissions": [
    {
      "id": 148,
      "action": "view",
      "resourceType": "dashboard",
      "resourceId": null
    },
    {
      "id": 58,
      "action": "myschedule",
      "resourceType": "taskscheduling",
      "resourceId": null
    }
  ],
}
```

```
{
  "id":59,
  "action":"managecredentials",
  "resourceType":"credentials",
  "resourceId":null
},
{
  "id":30,
  "action":"view",
  "resourceType":"devices",
  "resourceId":null
},
{
  "id":150,
  "action":"manage",
  "resourceType":"eventtriggers",
  "resourceId":null
},
{
  "id":149,
  "action":"view",
  "resourceType":"eventtriggers",
  "resourceId":null
},
{
  "id":131,
  "action":"managemytriggers",
  "resourceType":"eventtriggers",
  "resourceId":null
}
],
"principals":[
  {
    "id":3
  }
]
```

```

    ]
  }

```

Request Parameters

Parameter	Type	Required	Description
name	String	Yes	Name of the role.
description	String	No	Description of the role.
permissions	Array	No	An array of permissions that will be granted for the role. Each permission requires the mandatory parameters. For more details on the parameters, see below.
principals	Array	No	An array/collection of principals (users) who will be granted access with the role. For more information on the parameters, see below.

permission array parameters

Parameter	Type	Required	Description
id	Integer	No	The numeric value that uniquely identifies the permission.
action	String	No	The action the permission enables.
resourceId	String	No	The resource id to which the action belongs.
resourceType	Array	No	<p>The resource group to which the action belongs.</p> <p>Typically a user is given the role permission in conjunction with user management permission.</p> <p>Roles and permissions</p>

principals array parameters

Parameter	Type	Required	Description
id	Integer	No	Id of the user.
username	String	No	User name of the user.
subjectId	String	No	Subject Id of the user.
domain	String	No	Active directory domain, if the user is an AD User.
autoLoginEnabled	Boolean	No	Flag to indicate if auto login is enabled or not.
deleted	Boolean	No	Flag to indicate if user is deleted or not.
emailVerified	Boolean	No	Flag to indicate if email is verified or not.
pwdExpired	Boolean	No	Flag to indicate if password is expired or not.

Response

201 Created

```
{
  "id":25,
  "createdBy":1,
  "createdOn":"2022-04-11T11:53:03Z",
  "updatedBy":1,
  "updatedOn":"2022-04-11T11:53:03Z",
  "tenantId":1,
  "version":0,
  "tenantUuid":"282978c4-6386-c13a-92ac-5009e3cfd6b3",
```

```
"description":"View and Manage the triggers",
"name":"Trigger Manager",
"permissions":[
  {
    "id":59,
    "createdBy":0,
    "createdOn":"2022-02-28T23:49:21Z",
    "updatedBy":0,
    "updatedOn":"2022-02-28T23:49:21Z",
    "tenantId":1,
    "version":0,
    "tenantUuid":"282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "action":"managecredentials",
    "resourceId":null,
    "resourceType":"credentials"
  },
  {
    "id":131,
    "createdBy":0,
    "createdOn":"2022-02-28T23:49:31Z",
    "updatedBy":0,
    "updatedOn":"2022-02-28T23:49:31Z",
    "tenantId":1,
    "version":0,
    "tenantUuid":"282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "action":"managemytriggers",
    "resourceId":null,
    "resourceType":"eventtriggers"
  },
  {
    "id":149,
    "createdBy":0,
    "createdOn":"2022-02-28T23:49:42Z",
    "updatedBy":0,
    "updatedOn":"2022-02-28T23:49:42Z",
    "tenantId":1,
```

```

    "version":0,
    "tenantUuid":"282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "action":"view",
    "resourceId":null,
    "resourceType":"eventtriggers"
  },
  {
    "id":58,
    "createdBy":0,
    "createdOn":"2022-02-28T23:49:21Z",
    "updatedBy":0,
    "updatedOn":"2022-02-28T23:49:21Z",
    "tenantId":1,
    "version":0,
    "tenantUuid":"282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "action":"myschedule",
    "resourceId":null,
    "resourceType":"taskscheduling"
  },
  {
    "id":148,
    "createdBy":0,
    "createdOn":"2022-02-28T23:49:38Z",
    "updatedBy":0,
    "updatedOn":"2022-02-28T23:49:38Z",
    "tenantId":1,
    "version":0,
    "tenantUuid":"282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "action":"view",
    "resourceId":null,
    "resourceType":"dashboard"
  },
  {
    "id":150,
    "createdBy":0,
    "createdOn":"2022-02-28T23:49:42Z",

```

```
    "updatedBy":0,
    "updatedOn":"2022-02-28T23:49:42Z",
    "tenantId":1,
    "version":0,
    "tenantUuid":"282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "action":"manage",
    "resourceId":null,
    "resourceType":"eventtriggers"
  },
  {
    "id":30,
    "createdBy":0,
    "createdOn":"2022-02-28T23:49:21Z",
    "updatedBy":0,
    "updatedOn":"2022-02-28T23:49:21Z",
    "tenantId":1,
    "version":0,
    "tenantUuid":"282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "action":"view",
    "resourceId":null,
    "resourceType":"devices"
  }
],
"countPrincipals":0,
"systemRole":false,
"principals":[
  {
    "id":3,
    "createdBy":1,
    "createdOn":"2022-03-17T19:33:06Z",
    "updatedBy":1,
    "updatedOn":"2022-04-08T21:54:29Z",
    "tenantId":1,
    "version":274,
    "tenantUuid":"282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "username":"john_doe",
```

```

    "description": "",
    "deleted": false,
    "disabled": false,
    "email": "aa@aa.com",
    "firstName": "John",
    "lastName": "Doe",
    "autoLoginEnabled": true,
    "emailVerified": true,
    "clientRegistered": false,
    "passwordSet": true,
    "questionsSet": true,
    "activeDirectory": false,
    "passwordChangedOn": "2022-03-17T19:33:59Z",
    "deviceCredentialAttested": false,
    "multipleLoginAllowed": true
  }
]
}

```

Response Parameters

Parameter	Type	Description
id	Integer	Unique identifier representing the new role created.
name	String	Name of the role created.
description	String	Description of the role created.
version	Integer	Version of the role instance.
createdBy	Integer	Id of the user who created the role.
createdOn	String	The creation timestamp of the role.

Parameter	Type	Description
updatedBy	Integer	Id of the user who made a latest update to the role.
updatedAt	String	The latest update timestamp of the role.
permissions	Array	An array of unique permissions that have been assigned to the role.
principals	Array	An array of unique users that have been assigned to the role.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

List roles

Use the List Roles API to retrieve a list of roles in the Control Room. The endpoint supports pagination, sorting, and filtering.

Request

```
POST http://{{ControlRoomURL}}/v1/usermanagement/roles/list
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request body without filters:

```
{
  "sort": [
    {
      "field": "name",
```

```

        "direction": "asc"
      }
    ],
    "filter": {

    },
    "page": {
      "offset": 0,
      "total": 100,
      "totalFilter": 100,
      "length": 200
    }
  }
}

```

Request body with filters:

```

{
  "sort": [
    {
      "field": "name",
      "direction": "asc"
    }
  ],
  "filter": {
    "operator": "and",
    "operands": [
      {
        "operator": "substring",
        "field": "name",
        "value": "Device"
      },
      {
        "operator": "gt",
        "field": "createdOn",
        "value": "2022-02-01T00:00:00.989Z"
      }
    ]
  }
}

```

```

        "operator": "lt",
        "field": "createdOn",
        "value": "2022-03-20T23:00:00.123Z"
      }
    ]
  },
  "page": {
    "offset": 0,
    "total": 100,
    "totalFilter": 100,
    "length": 200
  }
}

```

Request Parameters

Parameter	Type	Required	Description
sort	Array	No	<p>By default, search results are sorted in descending order with respect to their ids. An alternative sorting is specified using the sort query parameter.</p> <p>Enter the field by which you want to sort along with the direction <code>asc</code> (ascending) or <code>desc</code> (descending).</p>
filter	Object	No	Filters the result.
page	Object	No	The page object allows you to get the desired pages.

For more information on Filtering, Pagination, and Sorting, see [Filtering, pagination, and sorting](#).

Response

```

{
  "page": {

```



```

    "offset": 0,
    "total": 21,
    "totalFilter": 1
  },
  "list": [
    {
      "id": 24,
      "name": "Device_admin",
      "description": "This is a device admin role",
      "countPrincipals": 1,
      "version": 1,
      "createdBy": 1,
      "createdOn": "2022-03-17T19:32:20.620Z",
      "updatedBy": 1,
      "updatedOn": "2022-03-24T02:20:13.787Z",
      "systemRole": false
    }
  ]
}

```

Response Parameters

Parameter	Type	Description
offset	Integer	The starting list offset, used for pagination.
total	Integer	Total number of records.
totalFilter	Integer	Number of records after applying the filter.
List	Array	The array of List roles object.
List roles object		
id	Integer	The unique Id of a specific role.

Parameter	Type	Description
name	String	Name of role.
description	String	Description of role.
countPrincipals	Integer	Count of Principals (users) who are granted with this role.
version	Integer	Version of the role instance.
createdBy	Integer	Id of the user who created the role.
createdOn	String	The creation timestamp of the role.
updatedBy	Integer	Id of the user who made a latest update to the role.
updatedOn	String	The latest update timestamp of the role.
systemRole	Boolean	Flag to indicate if this role is system role or not.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Retrieve role

Use the Get role by ID API to retrieve a specific role in the Control Room.

Request

```
GET https://{ControlRoomURL}/v1/usermanagement/roles/<role ID>
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request Parameters

Parameter	Type	Required	Description
role ID	Integer	Yes	Enter the Id of the role.

Response

200 OK

```
{
  "id": 264,
  "createdBy": 10,
  "createdOn": "2022-04-12T12:12:39Z",
  "updatedBy": 10,
  "updatedOn": "2022-04-12T12:12:39Z",
  "tenantId": 1,
  "version": 0,
  "tenantUuid": "06e42523-b44a-49f4-82dc-b8d420896761",
  "description": "",
  "name": "Device admin",
  "permissions": [
    {
      "id": 1345,
      "createdBy": 0,
      "createdOn": "2022-04-11T19:46:24Z",
      "updatedBy": 1,
      "updatedOn": "2022-04-11T19:46:24Z",
      "tenantId": 1,
      "version": 0,
```

```
    "tenantUuid": "06e42523-b44a-49f4-82dc-b8d420896761",
    "action": "managecredentials",
    "resourceId": null,
    "resourceType": "credentials"
  },
  {
    "id": 1424,
    "createdBy": 0,
    "createdOn": "2022-04-11T19:46:24Z",
    "updatedBy": 1,
    "updatedOn": "2022-04-11T19:46:24Z",
    "tenantId": 1,
    "version": 0,
    "tenantUuid": "06e42523-b44a-49f4-82dc-b8d420896761",
    "action": "delete",
    "resourceId": null,
    "resourceType": "devices"
  },
  {
    "id": 1425,
    "createdBy": 0,
    "createdOn": "2022-04-11T19:46:24Z",
    "updatedBy": 1,
    "updatedOn": "2022-04-11T19:46:24Z",
    "tenantId": 1,
    "version": 0,
    "tenantUuid": "06e42523-b44a-49f4-82dc-b8d420896761",
    "action": "edit",
    "resourceId": null,
    "resourceType": "devices"
  },
  {
    "id": 1344,
    "createdBy": 0,
    "createdOn": "2022-04-11T19:46:24Z",
    "updatedBy": 1,
```

```

    "updatedOn": "2022-04-11T19:46:24Z",
    "tenantId": 1,
    "version": 0,
    "tenantUuid": "06e42523-b44a-49f4-82dc-b8d420896761",
    "action": "myschedule",
    "resourceId": null,
    "resourceType": "taskscheduling"
  },
  {
    "id": 1426,
    "createdBy": 0,
    "createdOn": "2022-04-11T19:46:24Z",
    "updatedBy": 1,
    "updatedOn": "2022-04-11T19:46:24Z",
    "tenantId": 1,
    "version": 0,
    "tenantUuid": "06e42523-b44a-49f4-82dc-b8d420896761",
    "action": "view",
    "resourceId": null,
    "resourceType": "dashboard"
  },
  {
    "id": 1414,
    "createdBy": 0,
    "createdOn": "2022-04-11T19:46:24Z",
    "updatedBy": 1,
    "updatedOn": "2022-04-11T19:46:24Z",
    "tenantId": 1,
    "version": 0,
    "tenantUuid": "06e42523-b44a-49f4-82dc-b8d420896761",
    "action": "attestcredentials",
    "resourceId": null,
    "resourceType": "devices"
  },
  {
    "id": 1381,

```

```
"createdBy": 0,  
"createdOn": "2022-04-11T19:46:24Z",  
"updatedBy": 1,  
"updatedOn": "2022-04-11T19:46:24Z",  
"tenantId": 1,  
"version": 0,  
"tenantUuid": "06e42523-b44a-49f4-82dc-b8d420896761",  
"action": "register",  
"resourceId": null,  
"resourceType": "devices"  
},  
{  
  "id": 1321,  
  "createdBy": 0,  
  "createdOn": "2022-04-11T19:46:24Z",  
  "updatedBy": 1,  
  "updatedOn": "2022-04-11T19:46:24Z",  
  "tenantId": 1,  
  "version": 0,  
  "tenantUuid": "06e42523-b44a-49f4-82dc-b8d420896761",  
  "action": "view",  
  "resourceId": null,  
  "resourceType": "devices"  
},  
{  
  "id": 1423,  
  "createdBy": 0,  
  "createdOn": "2022-04-11T19:46:24Z",  
  "updatedBy": 1,  
  "updatedOn": "2022-04-11T19:46:24Z",  
  "tenantId": 1,  
  "version": 0,  
  "tenantUuid": "06e42523-b44a-49f4-82dc-b8d420896761",  
  "action": "all",  
  "resourceId": null,  
  "resourceType": "devices"
```

```
    }  
  ],  
  "countPrincipals": 0,  
  "systemRole": false,  
  "principals": [  
    {  
      "id": 21,  
      "createdBy": 10,  
      "createdOn": "2022-04-12T12:12:19Z",  
      "updatedBy": 21,  
      "updatedOn": "2022-04-13T03:37:08Z",  
      "tenantId": 1,  
      "version": 19,  
      "tenantUuid": "06e42523-b44a-49f4-82dc-b8d420896761",  
      "username": "john_user",  
      "description": "User",  
      "deleted": false,  
      "disabled": false,  
      "email": "john.doe@aa.com",  
      "firstName": "John",  
      "lastName": "Doe",  
      "autoLoginEnabled": true,  
      "emailVerified": true,  
      "clientRegistered": false,  
      "passwordSet": true,  
      "questionsSet": true,  
      "activeDirectory": false,  
      "passwordChangedOn": "2022-04-12T12:13:35Z",  
      "deviceCredentialAttested": false,  
      "multipleLoginAllowed": true  
    }  
  ]  
}
```

Response Parameters

Parameter	Type	Description
id	Integer	Unique identifier representing the new role created.
name	String	Name of the role created.
description	String	Description of the role created.
version	Integer	Version of the role instance.
createdBy	Integer	Id of the user who created the role.
createdOn	String	The creation timestamp of the role.
updatedBy	Integer	Id of the user who made a latest update to the role.
updatedOn	String	The latest update timestamp of the role.
permissions	Array	An array of unique permissions that have been assigned to the role.
principals	Array	An array of unique users that have been assigned to the role.

 **Note:** You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Update role

Use the Update role API to update an existing role in the Control Room.

Request

```
PUT https://{ControlRoomURL}/v1/usermanagement/roles/<role ID>
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request body:

```
{
  "id":25,
  "name":"Trigger Manager",
  "principals":[
    {
      "id":3
    },
    {
      "id":4
    }
  ],
  "description":"View and Manage the triggers",
  "permissions":[
    {
      "id":148,
      "action":"view",
      "resourceType":"dashboard",
      "resourceId":null
    },
    {
      "id":58,
      "action":"myschedule",
      "resourceType":"taskscheduling",
      "resourceId":null
    },
    {
```

```
    "id":149,
    "action":"view",
    "resourceType":"eventtriggers",
    "resourceId":null
  },
  {
    "id":150,
    "action":"manage",
    "resourceType":"eventtriggers",
    "resourceId":null
  },
  {
    "id":131,
    "action":"managemytriggers",
    "resourceType":"eventtriggers",
    "resourceId":null
  },
  {
    "id":59,
    "action":"managecredentials",
    "resourceType":"credentials",
    "resourceId":null
  },
  {
    "id":30,
    "action":"view",
    "resourceType":"devices",
    "resourceId":null
  }
],
"existingRepositoryPermissions":[

],
"version":0
}
```

Request Parameters

Parameter	Type	Required	Description
name	String	Yes	Enter the name of the role.
description	String	No	Description of the role.
permissions	Array	No	An array of permissions that will be granted for the role. Each permission requires the mandatory parameters. For more details on the parameters, see below.
principals	Array	No	An array/collection of principals (users) who will be granted access with the role. For more information on the parameters, see below.

permission array parameters

Parameter	Type	Required	Description
id	Integer	No	The numeric value that uniquely identifies the permission.
action	String	No	The action the permission enables.
resourceId	String	No	The resource id to which the action belongs.
resourceType	Array	No	The resource group to which the action belongs. Typically a user is given the role permission in conjunction with user management permission. Roles and permissions

principals array parameters

Parameter	Type	Required	Description
id	Integer	No	Id of the user.
username	String	No	User name of the user.
subjectId	String	No	Subject Id of the user.
domain	String	No	Active directory domain, if the user is an AD User.
autoLoginEnabled	Boolean	No	Flag to indicate if auto login is enabled or not.
deleted	Boolean	No	Flag to indicate if user is deleted or not.
emailVerified	Boolean	No	Flag to indicate if email is verified or not.
pwdExpired	Boolean	No	Flag to indicate if password is expired or not.

Response

200 OK

```
{
  "id":25,
  "createdBy":1,
  "createdOn":"2022-04-11T11:53:03Z",
  "updatedBy":1,
  "updatedOn":"2022-04-11T12:01:31Z",
  "tenantId":1,
  "version":1,
  "tenantUuid":"282978c4-6386-c13a-92ac-5009e3cfd6b3",
  "description":"View and Manage the triggers",
```

```
"name": "Trigger Manager",
"permissions": [
  {
    "id": 59,
    "createdBy": 0,
    "createdOn": "2022-02-28T23:49:21Z",
    "updatedBy": 0,
    "updatedOn": "2022-02-28T23:49:21Z",
    "tenantId": 1,
    "version": 0,
    "tenantUuid": "282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "action": "managecredentials",
    "resourceId": null,
    "resourceType": "credentials"
  },
  {
    "id": 131,
    "createdBy": 0,
    "createdOn": "2022-02-28T23:49:31Z",
    "updatedBy": 0,
    "updatedOn": "2022-02-28T23:49:31Z",
    "tenantId": 1,
    "version": 0,
    "tenantUuid": "282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "action": "managemytriggers",
    "resourceId": null,
    "resourceType": "eventtriggers"
  },
  {
    "id": 149,
    "createdBy": 0,
    "createdOn": "2022-02-28T23:49:42Z",
    "updatedBy": 0,
    "updatedOn": "2022-02-28T23:49:42Z",
    "tenantId": 1,
    "version": 0,
```

```

    "tenantUuid":"282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "action":"view",
    "resourceId":null,
    "resourceType":"eventtriggers"
  },
  {
    "id":58,
    "createdBy":0,
    "createdOn":"2022-02-28T23:49:21Z",
    "updatedBy":0,
    "updatedOn":"2022-02-28T23:49:21Z",
    "tenantId":1,
    "version":0,
    "tenantUuid":"282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "action":"myschedule",
    "resourceId":null,
    "resourceType":"taskscheduling"
  },
  {
    "id":148,
    "createdBy":0,
    "createdOn":"2022-02-28T23:49:38Z",
    "updatedBy":0,
    "updatedOn":"2022-02-28T23:49:38Z",
    "tenantId":1,
    "version":0,
    "tenantUuid":"282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "action":"view",
    "resourceId":null,
    "resourceType":"dashboard"
  },
  {
    "id":150,
    "createdBy":0,
    "createdOn":"2022-02-28T23:49:42Z",
    "updatedBy":0,

```

```

    "updatedOn": "2022-02-28T23:49:42Z",
    "tenantId": 1,
    "version": 0,
    "tenantUuid": "282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "action": "manage",
    "resourceId": null,
    "resourceType": "eventtriggers"
  },
  {
    "id": 30,
    "createdBy": 0,
    "createdOn": "2022-02-28T23:49:21Z",
    "updatedBy": 0,
    "updatedOn": "2022-02-28T23:49:21Z",
    "tenantId": 1,
    "version": 0,
    "tenantUuid": "282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "action": "view",
    "resourceId": null,
    "resourceType": "devices"
  }
],
"countPrincipals": 0,
"systemRole": false,
"principals": [
  {
    "id": 3,
    "createdBy": 1,
    "createdOn": "2022-03-17T19:33:06Z",
    "updatedBy": 1,
    "updatedOn": "2022-04-08T21:54:29Z",
    "tenantId": 1,
    "version": 274,
    "tenantUuid": "282978c4-6386-c13a-92ac-5009e3cfd6b3",
    "username": "john_user",
    "description": ""
  }
]

```

```
"deleted":false,
"disabled":false,
"email":"john.doe@aa.com",
"firstName":"John",
"lastName":"Doe",
"autoLoginEnabled":true,
"emailVerified":true,
"clientRegistered":false,
"passwordSet":true,
"questionsSet":true,
"activeDirectory":false,
"passwordChangedOn":"2022-03-17T19:33:59Z",
"deviceCredentialAttested":false,
"multipleLoginAllowed":true
},
{
  "id":4,
  "createdBy":1,
  "createdOn":"2022-04-04T15:32:38Z",
  "updatedBy":1,
  "updatedOn":"2022-04-04T15:32:38Z",
  "tenantId":1,
  "version":11,
  "tenantUuid":"282978c4-6386-c13a-92ac-5009e3cfd6b3",
  "username":"test_admin",
  "description":"",
  "deleted":false,
  "disabled":false,
  "email":"testadmin@aa.com",
  "firstName":"Test",
  "lastName":"Admin",
  "autoLoginEnabled":false,
  "emailVerified":true,
  "clientRegistered":false,
  "passwordSet":true,
  "questionsSet":true,
```



```

    "activeDirectory":false,
    "passwordChangedOn":"2022-04-04T15:33:22Z",
    "deviceCredentialAttested":false,
    "multipleLoginAllowed":true
  }
]
}

```

Response Parameters

Parameter	Type	Description
id	Integer	Unique identifier representing the new role created.
name	String	Name of the role created.
description	String	Description of the role created.
version	Integer	Version of the role instance.
createdBy	Integer	Id of the user who created the role.
createdOn	String	The creation timestamp of the role.
updatedBy	Integer	Id of the user who made a latest update to the role.
updatedOn	String	The latest update timestamp of the role.
permissions	Array	An array of unique permissions that have been assigned to the role.
principals	Array	An array of unique users that have been assigned to the role.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Delete role

Use the Delete role API to delete an existing role in the Control Room.

Request

```
DELETE https://{ControlRoomURL}/v1/usermanagement/roles/<role ID>
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request Parameters

Parameter	Type	Required	Description
role ID	Integer	Yes	Enter the Id of the role.

Response

```
200 OK
```

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Create user

Use the Create user API to create a new user in the Control Room.

Request

```
POST https://{ControlRoomURL}/  
v1/usermanagement/users
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<b  
earer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request body:

```
{  
  "roles": [  
    {  
      "id": 185  
    }  
  ],  
  "email": "aaa@a.com",  
  "enableAutoLogin": true,  
  "username": "test1",  
  "description": "",  
  "firstName": "joe",  
  "lastName": "doe",  
  "disabled": false,  
  "password": "aa360aa360",  
  "licenseFeatures": [  
    "DEVELOPMENT"  
  ],  
  "sysAssignedRoles": [],  
  "deviceCredentialAttested": false  
}
```

Request Parameters

Parameter	Type	Required	Description
<code>roles</code>	Integer	Yes	Id of the role. To find an ID of a role, see List roles .
<code>domain</code>	string	No	ActiveDirectory domain
<code>email</code>	String	No	Email of the user
<code>enableAutoLogin</code>	Boolean	No	Flag to enable or disable Auto login.
<code>username</code>	String	Yes	Username of the user.
<code>password</code>	String	Yes	Password of the user.
<code>firstName</code>	String	No	First name of the user
<code>lastName</code>	String	No	Last name of the user
<code>disabled</code>	boolean	No	Enable or disable the user
<code>description</code>	String	Yes	Provide a description.
<code>licenseFeatures</code>	String	No	<p>License allocated to the user. You will be able to retrieve using List Control Room licenses.</p> <div> <p>Note: <code>licenseFeatures</code> is case sensitive in ALL CAPS. Valid <code>licenseFeatures</code> inputs include DEVELOPMENT, RUNTIME, ANALYTICSCLIENT, DISCOVERYBOTANALYZER, DISCOVERYBOT, AARIUSER, CITIZENDEVELOPER, and CLOUD.</p> </div>
<code>sysAssignedRoles</code>	Integer	No	System assigned roles

Parameter	Type	Required	Description
<code>deviceCredentialAttested</code>	String	No	Device Credential attested.

Response

201 Created

```
{
  "id": 129,
  "username": "test1",
  "domain": null,
  "firstName": "joe",
  "lastName": "doe",
  "version": 0,
  "principalId": 129,
  "deleted": false,
  "roles": [
    {
      "name": "AAE_Basic",
      "id": 185,
      "version": 1
    }
  ],
  "sysAssignedRoles": [],
  "groupNames": [],
  "permissions": [
    {
      "id": 1381,
      "action": "register",
      "resourceId": null,
      "resourceType": "devices"
    },
    {
      "id": 1346,
```

```
    "action": "createstandard",
    "resourceId": null,
    "resourceType": "credentialattribute"
  },
  {
    "id": 1378,
    "action": "view",
    "resourceId": null,
    "resourceType": "botstore"
  },
  {
    "id": 1417,
    "action": "viewuserrolebasicinfo",
    "resourceId": null,
    "resourceType": "usermanagement"
  },
  {
    "id": 1426,
    "action": "view",
    "resourceId": null,
    "resourceType": "dashboard"
  },
  {
    "id": 1376,
    "action": "view",
    "resourceId": null,
    "resourceType": "packagemanager"
  },
  {
    "id": 1305,
    "action": "run",
    "resourceId": null,
    "resourceType": "repositorymanager"
  },
  {
    "id": 1320,
```

```

        "action": "view",
        "resourceId": null,
        "resourceType": "repositorymanager"
    },
],
"licenseFeatures": [
    "DEVELOPMENT"
],
"emailVerified": true,
"passwordSet": false,
"questionsSet": false,
"enableAutoLogin": true,
"disabled": false,
"clientRegistered": false,
"description": "",
"createdBy": 21,
"createdOn": "2022-10-16T17:52:14Z",
"updatedBy": 21,
"updatedOn": "2022-10-16T17:52:14Z",
"publicKey": null,
"appType": null,
"routingName": null,
"appUrl": null,
"email": "aaa@a.com",
"lastLoginTime": null,
"deviceCredentialAttested": false,
"multipleLoginAllowed": false
}

```

Response Parameters

Parameter	Type	Description
id	Integer	Unique identifier representing the new user created.
Roles		

Parameter	Type	Description
name	String	Name of the role.
id	Integer	Id of the role
version	Integer	Version of the role instance.
Permissions		
permissions	Array	An array of unique permissions that have been assigned to the user.
id	Integer	Id of the permission.
action	String	Action associated with the permission.
resourceId	Integer	Resource Id
resourceType	String	Type of resource
createdBy	Integer	Id of the user who created the user.
createdOn	String	The creation timestamp of the user.
updatedBy	Integer	Id of the user who made a latest update to the user.
updatedOn	String	The latest update timestamp of the user.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Search for users API

Use the Search for users API to search for all users in the Control Room.

Request

```
POST http://{{ControlRoomURL}}/v1/usermanagement/users/list
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

To search for users, you must have one of the following:

- **Admin** role
- Custom role with **View Users** permission

Request body:

```
{
  "filter": {
    "operator": "and",
    "operands": [
      "substring"
    ],
    "field": "name",
    "value": "device"
  },
  "sort": [
    {
      "field": "name",
      "direction": "asc"
    }
  ],
  "page": {
```

```

    "offset": 0,
    "total": 100,
    "totalFilter": 100
  }
}

```

Request Parameters

Parameter	Type	Required	Description
sort	Array	No	<p>By default, search results are sorted in descending order of the IDs. To specify an alternate sorting, use the sort query parameter.</p> <p>Enter the field by which you want to sort along with the direction <code>asc</code> (ascending) or <code>desc</code> (descending).</p>
filter	Object	No	Filters the result.
page	Object	No	The page object allows you to get the desired number of pages.

For more information on Filtering, Pagination, and Sorting, see [Filtering, pagination, and sorting](#).

Response

```

{
  {
    "page": {
      "offset": 0,
      "total": 100,
      "totalFilter": 100
    },
    "list": [
      {
        "id": 110,

```

```
"roles": [  
  {  
    "id": 0,  
    "name": "string"  
  }  
],  
"permissions": [  
  {  
    "id": 59,  
    "action": "managecredentials",  
    "resourceId": null,  
    "resourceType": "credentials"  
  }  
],  
"licenseFeatures": [  
  "DEVELOPMENT"  
],  
"principalId": 110,  
"domain": null,  
"email": "xyz@automationanywhere.com",  
"emailVerified": true,  
"passwordSet": true,  
"questionsSet": true,  
"enableAutoLogin": true,  
"username": "bot_creator",  
"firstName": "Mike",  
"lastName": "Bots",  
"description": "User to manage bots",  
"disabled": true,  
"clientRegistered": true,  
"createdBy": 27,  
"createdOn": "2022-03-17T19:32:20.620Z",  
"updatedBy": 29,  
"updatedOn": "2022-03-24T02:20:13.787Z"  
}
```

```

    ]
  }

```

Response Parameters

Parameter	Type	Description
offset	Integer	The starting list offset, used for pagination.
total	Integer	Total number of records.
totalFilter	Integer	Number of records after applying the filter.
List	Array	The array of List roles object.
List roles object		
id	Integer	The unique Id of a specific role.
name	String	Name of role.
permissions	Array	Collection of unique permissions that have been assigned to the role.
licenseFeatures	String	License features that include - DEVELOPMENT - RUNTIME - IQBOTRUNTIME - ANALYTICSCLIENT - ANALYTICSAPI
description	String	Description of the role.
principalId	Number	Unique ID of the principal user.
email	Integer	Email ID of the user.
emailVerified	Boolean	Indicates if the email ID has been verified.

Parameter	Type	Description
passwordSet	Boolean	Indicates if the user had set his password.
enableAutoLogin	Boolean	Indicates if auto login is enabled for the user.
username	String	User log in name.
firstName	String	First name specified for the user.
lastName	String	Last name specified for the user.
createdBy	Integer	ID of the user who created the role.
createdOn	String	Timestamp when the role was created.
updatedBy	Integer	ID of the user who last updated the role.
updatedOn	String	Timestamp when the role was last updated.
systemRole	Boolean	Indicates if this role is a system role or not.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Retrieve a specific user details API

Use the Get user details API to retrieve a specific user details in the Control Room.

Request

```
GET http://{{ControlRoomURL}}/v1/usermanagement/users/{uid}
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Ensure you have an admin role or a custom role with **View Users** permission.

Request Parameters

Parameter	Type	Required	Description
uid	Integer	Yes	Enter the unique ID of the user.

Response

```
{
  "id": 110,
  "roles": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "permissions": [
    {
      "id": 59,
      "action": "managecredentials",
      "resourceId": null,
      "resourceType": "credentials"
    }
  ],
  "licenseFeatures": [
    "DEVELOPMENT"
  ],
  "principalId": 110,
  "domain": null,
```

```

"email": "xyz@automationanywhere.com",
"emailVerified": true,
"passwordSet": true,
"questionsSet": true,
"enableAutoLogin": true,
"username": "bot_creator",
"firstName": "Mike",
"lastName": "Bots",
"description": "User to manage bots",
"disabled": true,
"clientRegistered": true,
"createdBy": 27,
"createdOn": "2022-03-17T19-32-20",
"updatedBy": 29,
"updatedOn": "2022-03-24T02-20-13"
}

```

Response Parameters

Parameter	Type	Description
id	Integer	Unique identifier representing a specific permission
roles		
id	Integer	Unique identifier representing a specific role
name	String	Name of role.
permissions	Array	Collection of unique permissions that have been assigned to the role.
licenseFeatures	String	License features that include - DEVELOPMENT - RUNTIME - IQBOTRUNTIME - ANALYTICSCLIENT - ANALYTICSAPI
description	String	Description of the role.

Parameter	Type	Description
domain	String	ActiveDirectory(LDAP) domain
principalId	Number	Unique ID of the principal user.
email	Integer	Email ID of the user.
emailVerified	Boolean	Indicates if the email ID has been verified.
passwordSet	Boolean	Indicates if the user had set his password.
questionsSet	Boolean	Indicates if the user has set questions and answer.
enableAutoLogin	Boolean	Indicates if auto login is enabled for the user.
firstName	String	First name specified for the user.
lastName	String	Last name specified for the user.
disabled	Boolean	User enable/disable flag.
clientRegistered	Boolean	Flag to indicate if client/device is registered.
createdBy	Integer	ID of the user who created the role.
createdOn	String	Timestamp when the role was created.
updatedBy	Integer	ID of the user who last updated the role.
updatedOn	String	Timestamp when the role was last updated.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Update an existing user details API

Use the Update user details API to update an existing user information in the Control Room.

Prerequisites

Ensure you have an admin role or a custom role with **Edit Users** permission.

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the PUT method and endpoint URL: `<your_control_room_url>/v1/usermanagement/users/744`
3. In the request header, add an existing user ID you want to update. To find a user ID you want to update, execute the Search for users API.

If you want to add a new role ID to your request, perform the following steps:

- a. Execute the Search for users API. Use the POST method and endpoint URL: `<your_control_room_url>/v1/usermanagement/users/list`
- b. When you get all role IDs, add a new role to the existing role IDs. You will not be able to add one role by itself, you must add it to the collection of role IDs.

The following request body is for an existing user ID: 744 and the existing role IDs: 169, 2, and 26. Modify other parameters as needed.

Request body

```
{
  "roles": [
```

```
{
  "id": 169
},
{
  "id": 2
},
{
  "id": 26
}
],
"email": "Joe.Smith@automationanywhere.com",
"enableAutoLogin": false,
"firstName": "FN",
"lastName": "LN",
"description": "test",
"disabled": false,
"licenseFeatures": [
  "RUNTIME"
]
}
```

4. Send the request.

The response body returns the updated details for the user ID: 744.

Response body:

```
{
  "id": 744,
  "username": "cs_runner",
  "domain": null,
  "firstName": "FN",
  "lastName": "LN",
  "version": 60,
  "principalId": 744,
  "deleted": false,
  "roles": [
    {
```

```
        "name": "AAE_Basic",
        "id": 2,
        "version": 0
    },
    {
        "name": "cs_role2",
        "id": 169,
        "version": 3
    },
    {
        "name": "all",
        "id": 26,
        "version": 54
    }
],
"sysAssignedRoles": [],
"groupNames": [],
"permissions": [
    {
        "id": 2912,
        "action": "upload",
        "resourceId": "34241",
        "resourceType": "repositorymanager"
    },
    .....
    {
        "id": 4101,
        "action": "download",
        "resourceId": "34439",
        "resourceType": "repositorymanager"
    }
],
"licenseFeatures": [
    "RUNTIME"
],
"emailVerified": true,
```

```
"passwordSet": true,
"questionsSet": true,
"enableAutoLogin": false,
"disabled": false,
"clientRegistered": false,
"description": "test",
"createdBy": 451,
"createdOn": "2020-08-25T07:27:58Z",
"updatedBy": 451,
"updatedOn": "2021-03-16T17:15:19Z",
"publicKey": null,
"appType": null,
"routingName": null,
"appUrl": null,
"email": "Joe.Smith@automationanywhere.com",
"lastLoginTime": "2021-02-25T18:01:40Z",
"deviceCredentialAttested": false
}
```

The REST API responds to each request with an HTTP response code. For details about the response codes, see [API response codes](#).

Delete an existing user API

Use the Delete user API to delete an existing user in the Control Room.

Request

```
DELETE https://{ControlRoomURL}/v1/usermanagement/users/<user ID>
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request Parameters

Parameter	Type	Required	Description
user ID	Integer	Yes	Enter the Id of the user.

Response

200 OK

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Prerequisites

Ensure you have an admin role or a custom role with **Edit Users** permission.

- URL: `http://<your_control_room_url>/v1/usermanagement/users/2 <user ID>`

Replace the content in the angle brackets with your Control Room URL.

- Method: **DELETE**

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. In the request header, add an existing user ID you want to delete.
3. Use the **DELETE** method. and endpoint URL: `<your_control_room_url>/v1/usermanagement/users/2 <user ID>`
4. Send the request.

Response body:

```
{
  "id": 3014,
  "email": "a@a.com",
  "username": "docstest01",
```

```
"domain": null,
"firstName": null,
"lastName": null,
"version": 4,
"principalId": 3014,
"deleted": false,
"roles": [],
"sysAssignedRoles": [],
"groupNames": [],
"permissions": [],
"licenseFeatures": [],
"emailVerified": true,
"passwordSet": false,
"questionsSet": false,
"enableAutoLogin": false,
"disabled": false,
"clientRegistered": false,
"description": null,
"createdBy": 2623,
"createdOn": "2020-01-31T17:33:16Z",
"updatedBy": 3215,
"updatedOn": "2020-03-22T22:51:48Z",
"publicKey": null,
"appType": null,
"routingName": null,
"appUrl": null
}
```

The REST API responds to each request with an HTTP response code. For details about the response codes, see [API response codes](#).

Roles and permissions

Assign roles from the Automation 360 Administration user interface or through the User Management API to enable users to access features. You can assign a system-created role or create a custom role with specific permissions.

Roles are a logical container for permissions and have interdependencies with bots, users, and licenses. Users with the `AAE_Admin` role can create custom roles and assign roles to users.

The following topics provide descriptions of the features and the necessary information to create and assign roles with the [User Management API](#).

- **[System-created roles](#)**

Automation 360 includes predefined roles that cannot be edited or deleted. Each role has the standard permissions, plus the permissions necessary to perform the tasks within the scope of that role.

- **[Administration permissions](#)**

Enable users to create and manage users and roles, to manage and update migrations, and to install Control Room licenses.

- **[API permissions](#)**

Enables access to Bot Insight APIs and the ability to generate an API key.

- **[Audit log permissions](#)**

View logs and details to record user activities. Enable users to view logs from the Control Room.

- **[Automation Anywhere Robotic Interface \(AARI\) permissions](#)**

Assign AARI permissions to a custom role.

- **[Bot and bot credential permissions](#)**

Enables users to access features for managing bots and the credentials used by bots.

- **[Bot Store permissions](#)**

Enables users to view and manage Bot Store packages.

- **[Dashboard and activity permissions](#)**

Enables all users to view dashboards. Activity permissions enable users to view, manage, and schedule bot activities.

- **[Devices permissions](#)**

Enable you to register, view, and manage devices used to run bots.

- **[Discovery Bot permissions](#)**

Enable users to view or manage Discovery Bot processes, recordings, aggregations, and opportunities.

- **[Event triggers permissions](#)**

Enable users to run bots automatically depending on a specific event, such as a new window opening. You can limit users ability to only view or to view and manage triggers.

- **[IQ Bot permissions](#)**

Enable users to view IQ Bot permissions, create and manage learning instances, import and export domains, and manage IQ Bot configuration settings and migration.

- **[MetaBot permission](#)**

MetaBots are obsolete in Automation 360. However, this permission is valid to ensure that all functionality previously available is supported in Automation 360.

- **[Package manager permissions](#)**

Enable users to view and manage packages.

- **[Workload permissions](#)**

Enable you to create and manage workitems, workitem models, queues, and automations in the Control Room.

Parent topic: [User management APIs](#)

System-created roles

Automation 360 includes predefined roles that cannot be edited or deleted. Each role has the standard permissions, plus the permissions necessary to perform the tasks within the scope of that role.

Standard permissions

All system-created roles include the following permissions:


- View dashboards
- View my in progress activity
- Manage my credentials and lockers
- Create standard attributes for a credential (except for Discovery Bot roles)
- View and manage my Bot runners, Bot creators, and devices
- View and manage my queues
- View users and roles basic information


System-created roles

Find the role IDs using the [List roles](#) and assign the role to a user in the [Create user](#) or [Update an existing user details API](#) endpoints.

Note: Since the role IDs vary for each cloud Control Room, you must look up the role ID by navigating to **Administration > Roles** and clicking the role name. The URL is `https://<control_room>/#/admin/roles/allroles/2/edit`, where 2 is the role name.

Name	Description
AAE_Admin	Allows access to all features, including creating other Admin users and access to all folders and files. The only role that can access Control Room settings.
AAE_Basic	Permissions to create credentials and set a standard attribute value, view and run their bots, view the Bot Store, register a device, and view packages.

Name	Description
AAE_Locker Admin	Can view all credentials and all lockers. They can change the owner of a credential that they do not own. For lockers they do not own, they can delete the locker, edit permissions, and remove credentials.
AAE_IQ Bot Validator	For a Bot Runner with an IQ Bot license. Permissions to access the IQ Bot Validator screen. Limited access to Control Room features.
AAE_Bot Insight Consumer	When combined with an Analytics license, this role grants the user the ability to view data in Bot Insight.
AAE_Bot Insight Expert	When combined with an Analytics license, this role grants the user the ability to view and manage data in Bot Insight.
AAE_IQ Bot Services	Permissions to access the IQ Bot console. Limited access to Control Room features.
AAE_Queue Admin	Permissions to view and manage all queues.
AAE_Pool Admin	<p>Permissions to view and manage all device pools.</p> <div>  Note: This role does not grant permission to view bots. </div>

Name	Description
AAE_Bot Insight Admin	The only role that can use Bot Insight APIs to access the data logged by the Control Room, and by a task during Production runs.
AAE_IQ Bot Admin	Allows access to all IQ Bot features.
AAE_Bot Store Publisher	Permissions to submit bot package or Digital Worker to Bot Store.
AAE_Bot Developer	<p>Permissions view, run, and import their bots, create folders, manage packages, and download bots and Digital Workers from the Bot Store to their private workspace.</p> <div>  Note: This role does not grant permission to register a device, or check bots in or out of the public workspace. </div>
AAE_Discovery Bot Admin	Allows access to view all Discovery Bot processes. Manages the creation, deletion, and editing of processes.
AAE_Discovery Bot User	Allows access to view the assigned Discovery Bot process assigned to a user. Create and run the Discovery Bot recorder for assigned business processes. Permissions to view, edit, and delete a user's own recordings.

Name	Description
AAE_Discovery Bot Analyst	Allows access to view and edit all approved recordings from assigned users for a given process. Permissions for system generated aggregated view of recordings to view, create, edit, and delete views. Permissions to create, view, edit, and delete opportunities for assigned processes. Export the opportunity to a word document and convert to a bot.
AAE_Robotic Interface Admin	Allows access to the Control Room and AARI on the web.
AAE_Robotic Interface Manager	Allows access to AARI on the web.
AAE_Robotic Interface User	Allows access to AARI on the web.



Parent topic: [Roles and permissions](#)

Administration permissions


Enable users to create and manage users and roles, to manage and update migrations, and to install Control Room licenses.

You will be able to use the [Retrieve role](#) endpoint with the role ID to retrieve the permissions assigned to any system or user defined roles.

Users and roles permissions


Action	Resource Type	Description
usermanagement	usermanagement	<p>Allows you to only view all other users in the system. You cannot create, edit, or delete users.</p> <div>  Note: You must assign this permission before assigning the <code>createuser</code> <code>updateuser</code> , or <code>deleteuser</code> permission. </div>
deleteuser	usermanagement	Allows you to delete other users the Control Room.
createuser	usermanagement	Allows you to create new users in the Control Room.
updateuser	usermanagement	Allows you to edit all users in the system.
rolesview	rolesmanagement	<p>Users with this permission are able to view the roles in the Control Room.</p> <div>  Note: You must assign this permission before assigning the <code>rolesmanagement</code> permission. </div>
rolesmanagement	rolesmanagement	Allows you to view and manage all roles in the Control Room.
viewuserrolebasicinfo	usermanagement	Allows you to view basic information on users and roles.

Migration permissions

Action	Resource Type	Description
view	migration	<p>Allows you to view new migrations, but not run them</p> <div>  Note: You must assign this permission before assigning the manage migration permission </div>
manage	migration	Allows you to view and run new migrations
updatestatus	migration	Allows Bot Runner Run-as user to update the bot conversion status in the Control Room

Licenses permissions

Action	Resource Type	Description
licensemanagement	licensemanagement	Allows you to view the license details for the Control Room.
licenseinstall	licensemanagement	Allows you to install Automation 360 licenses for the Control Room.
licenseuserallocation	licensemanagement	Allows you to assign device licenses to other users.

 **Note:** Only a user with the **AAE_Admin** role has the ability to view and manage settings in the Control Room. See [System-created roles](#).

Runtime Client Management permissions

Action	Resource Type	Description
runtimeclientsmanagement	runtimeclientsmanagement	Allows you to use the device mentioned in the resourceId for deployment. This permission is assigned when you are assigned a default device.
accessresourceany	runtimeclientsmanagement	Allows you to use any device for deployment. This permission is currently granted for all users with <u>AAE_ADMIN</u> role. ¹

Global Values Permissions

Action	Resource Type	Description
manageuserscopevalues	globalvalues	Manage tenant level global values, given to AAE_Admin only. This cannot be given to any custom role at the moment. ¹
managetenantscopevalues	globalvalues	Not used, created for future purpose. ¹

other permissions

Action	Resource Type	Description
systemadmin	system	<p>This permission is given to AAE_Admin only.</p> <p>It is a system call and cannot be called manually.</p> <p>This permission is used to count the number of</p>

Action	Resource Type	Description
		<p>pages consumed against the entitled pages for the IQ bot application.</p> <p>It fetches a list of all the users with basic details and license information.¹</p>
view	settings	<p>View settings.</p> <div> <p>Note: You will be allowed to view settings only with the system-created Admin role to view Settings.</p> </div>
all	botrunners	<p>Allows you to use any runAsUser for deployment. This permission is currently granted for all users with <u>AAE_ADMIN</u> role.¹</p>
operationroom	operationroom	<p>Legacy, not used and will be removed from the future releases.¹</p>
manage	mfa	<p>Legacy, not used and will be removed from the future releases.¹</p>

Parent topic: [Roles and permissions](#)

¹ Not available in the UI and is seen only in the API response.

API permissions

Enables access to Bot Insight APIs and the ability to generate an API key.

Action	Resource Type	Description
botinsightapi	api	<p>Allows access to Bot Insight RESTful APIs to the data logged by the Control</p>

Action	Resource Type	Description
		Room and by a task during production runs.
generateapikey	api	<p>Generate an apiKey that can be used in the Authentication API.</p> <p>Authenticate (username and apiKey)</p> <div> <p>Note: Without the <code>generateapikey</code> permission, use APIs by authenticating using their username and password.</p> <p>Authenticate (username and password)</p> </div>
coadmin	botinsightapi	Reserved. Not used for now and will be used when COE dashboard is available in A360. Allows you to access the COE dashboard when it is made available in A360.
accessreportingapi	iqbotapi	Allows you to access reporting details via an API.
accessuploadanddownloadapi	iqbotapi	Allows you to upload a document and download the digitized result via an API.
accessnlapi	iqbotapi	Allows you to access natural language processing capabilities of the IQ Bot portal via an API.

Parent topic: [Roles and permissions](#)

Audit log permissions

View logs and details to record user activities. Enable users to view logs from the Control Room.




Action	Resource Type	Description
recentactivities	recentactivities	Allows you to view all audit log activity for the Control Room
archiveaudit	recentactivities	Allows you to archive all audit log activity for the Control Room

Parent topic: [Roles and permissions](#)

Automation Anywhere Robotic Interface (AARI) permissions

Assign AARI permissions to a custom role.

Action	Resource Type	Description
aaricrossteamread	aari	View all teams and see the team members
aaricrossprocessread	aari	View all processes and see team members and managers
aariteammanagement	aari	Create and view teams, edit team names, descriptions, and process tags. Add new teams, team members, and assigned processes
aaritaskmanagement	aari	Submit and view tasks from processes that are assigned to the team
aaricasemanagement	aari	Create and view requests from processes that are assigned to the team



Action	Resource Type	Description
aariglobalcasemanagement	aari	<p>View process requests and tasks.</p> <div>  Note: This permission does not grant the ability to create a request or submit a task </div>
aariglobalprocessmanagement	aari	<p>View checked-in public processes and assigned managers and teams, edit process tags, and assign managers and teams</p> <div>  Note: This permission does not grant the ability to create a process. </div>
aariglobalteammanagement	aari	<p>View teams and assigned users, edit team names and descriptions. Add new processes, managers and users to a team</p> <div>  Note: This permission does not grant the ability to create a team. </div>
aarischeduler	aari	Allows users to be used as AARI scheduler

Parent topic: [Roles and permissions](#)

Bot and bot credential permissions


Enables users to access features for managing bots and the credentials used by bots.

Bots permissions

Action	Resource Type	Description
view	repositorymanager	<p>Allows you to view the bots they created and bots that were assigned to them.</p> <div> Note: You must assign this permission before assigning any other bots permissions.</div>
run	repositorymanager	<p>Allows you to run the bots they created and bots that were assigned to them.</p>
export	repositorymanager	<p>Allows you to export bots and related bot dependencies for which they have download permission.</p>
import	repositorymanager	<p>Allows you to import bots and bot dependencies for which they have upload permission.</p>
createfolders	repositorymanager	<p>Allows users to create folders within the folders that they have access to.</p>
renamefolders	repositorymanager	<p>Allows you to rename the folders they have access to.</p> <div> Note: Only empty folders can be renamed.</div>

Action	Resource Type	Description
cancelcheckout	repositorymanager	Allows you to cancel bot checkout and unlock the file from the public repository.
forceunlock	repositorymanager	Allows you to unlock locked bots.
accessresourceany	repositorymanager	Allows you to get permissions to the resources. For example, <i>runtimeclientsmanagement</i> , <i>pool</i> , <i>queue</i> and so on.
setproductionversion	repositorymanager	Allows you to set the production version of bots.
all	repositorymanager	Allows you to get all the repository permissions.
gitrestore	repositorymanager	Allows you to get git restore permissions.

Credentials and lockers permissions

Action	Resource Type	Description
managecredentials	credentials	<p>Allows you to create, edit, and delete their own credentials. In addition, the user can interact with credentials from their assigned lockers.</p> <div>  Note: All roles have this permission by default. </div>

Action	Resource Type	Description
create	locker	Allows you to create and manage their own lockers.
consume	locker	Allows you to consume a locker locker
createstandard	credentialattribute	Allows you to create a standard attribute for a credential that is shared across all users of that credential .
updateany	credentialattributevalue	Allows you to view and edit their own masked attributes.
botautologinapi	credentialattributevalue	Allows you to automate the login process to run bots remotely.
addcredential	credentialmanager	Legacy, not used and will be removed from the future releases.
updatecredential	credentialmanager	
deletecredential	credentialmanager	

Note: A user with the [AAE_Locker Admin](#) role can view all credentials and lockers in the Control Room. See [System-created roles](#).

Parent topic: [Roles and permissions](#)

Bot Store permissions

Enables users to view and manage Bot Store packages.

Action	Resource Type	Description
view	botstore	Allows you to view Bot Store.

Action	Resource Type	Description
addfrom	botstore	Allows you to add bot packages from Bot Store to their Control Room private workspace.
submit	botstore	Allows you to submit bot packages to Bot Store.

Parent topic: [Roles and permissions](#)

Dashboard and activity permissions

Enables all users to view dashboards. Activity permissions enable users to view, manage, and schedule bot activities.

Dashboard and Activity Permissions

Action	Resource Type	Description
view	dashboard	View dashboard. Note: All roles have this permission by default.
myschedule	taskscheduling	All users can view their own activity. Note: All roles have this permission by default.
managemyschedule	taskscheduling	All users can pause, resume, or cancel their own activity and move their finished activities to history.
manageeveryoneschedule	taskscheduling	A user can monitor ongoing automations where a user has either


Action	Resource Type	Description
		run or schedule access on the respective bot. The user can monitor and manage ongoing automations.
view	taskscheduling	Users can see their scheduled bots regardless of which user scheduled the bot.
addschedule	taskscheduling	It requires permission to view and manage Bot Runners.
updateschedule	taskscheduling	Users can edit their scheduled bots, even if the bots are scheduled by a different user.
deleteschedule	taskscheduling	Users can delete schedules for any of their bots regardless of which users scheduled the bot.
manageallmyfolderschedules	taskscheduling	Users can view, edit, and delete all the schedules on the bot folders that the user has access to. This includes the schedules that the user created or schedules created by other users.
manageallschedules	taskscheduling	Users can view, edit, and delete all the schedules in the system. This includes the schedules that the user created or schedules created by other users.
setautomationpriority	taskscheduling	Users can set the automation priority to high from the default medium. Automations with high priority are deployed ahead of automations with medium and low priority.

Action	Resource Type	Description
everyoneschedule	taskscheduling	Users can monitor those ongoing automations where they have either run or schedule access on the respective TaskBot.

Parent topic: [Roles and permissions](#)

Devices permissions

Enable you to register, view, and manage devices used to run bots.

Action	Resource Type	Description
register	devices	Allows you to register a localhost device.
all	devices	Allows you to view and manage all devices in the Control Room.
delete	devices	Allows you to delete devices that they registered.
edit	devices	Allows you to edit the devices that they have permission to see.
view	devices	<p>Allows you to view and manage Bot Creators, Bot Runners, and device pools.</p> <div>  Note: All roles have this permission by default. </div>
attestcredentials	devices	Allows you to deploy the bot during a session on the user's device without system password.

Action	Resource Type	Description
create	pool	Allows you to create and manage their own device pools.

Note: A user with the `AAE_Pool Admin` role is able to manage all device pools in the Control Room. See [System-created roles](#).

Parent topic: [Roles and permissions](#)


Discovery Bot permissions

Enable users to view or manage Discovery Bot processes, recordings, aggregations, and opportunities.

Process permissions

Action	Resource Type	Description
viewprocess	processdiscovery	<p>Allows users to view assigned processes.</p> <p>Note: This is the standard permission. You must assign this permission before assigning any process discovery permissions.</p>
viewallprocess	processdiscovery	Allows users to view all the defined processes.
editprocess	processdiscovery	Allows users to create and edit processes.

Recording permissions


Action	Resource Type	Description
viewrecording	processdiscovery	<p>Allows users to view their own recording.</p> <div>  Note: You must assign this permission before assigning any of the permissions below. </div>
createrecording	processdiscovery	Allows users to run recorder and create recording.
editrecording	processdiscovery	Allows users to edit their own recording.
deleterecording	processdiscovery	Allows users to delete their own recording.
viewallrecording	processdiscovery	Allows users to view all recordings.
editallrecording	processdiscovery	Allows users to edit all recordings.

Aggregation permissions

Action	Resource Type	Description
viewmanualaggregation	processdiscovery	Allows users to view their own aggregations.
viewallaggregations	processdiscovery	Allows users to view all aggregations.
createdeletemanualaggregation	processdiscovery	Allows users to create and delete aggregations.

Action	Resource Type	Description
updatemanualaggregation	processdiscovery	Allows users to update aggregations.
viewsystemaggregation	processdiscovery	Allows users to view system aggregations.

Opportunity permissions

Action	Resource Type	Description
viewopportunity	processdiscovery	<p>Allows users to view opportunities within an assigned process.</p> <div>  Note: You must assign this permission before assigning any of the opportunity permissions. </div>
editopportunity	processdiscovery	Allows users to edit opportunities within an assigned process.
viewalloppportunity	processdiscovery	Allows users to view all the opportunities.
createdeleteopportunity	processdiscovery	Allows users to create and delete opportunities within the assigned process.
converttobot	processdiscovery	Allows users to convert an opportunity to a bot.
exportopportunity	processdiscovery	Allows users to export an opportunity.

Parent topic: [Roles and permissions](#)

Event triggers permissions

Enable users to run bots automatically depending on a specific event, such as a new window opening. You can limit users ability to only view or to view and manage triggers.

Action	Resource type	Description
view	eventtriggers	Allows users to view event triggers.
manage	eventtriggers	Allows users to view and manage event triggers.
managemytriggers	eventtriggers	Allows users to view and manage event triggers from their private repository, on their local devices.

Additionally, you must enable the view packages permission as well:

Action	Resource type	Description
view	packagemanager	Allows users to view packages


Parent topic: [Roles and permissions](#)

IQ Bot permissions


Enable users to view IQ Bot permissions, create and manage learning instances, import and export domains, and manage IQ Bot configuration settings and migration.

IQ Bot permissions

Action	Resource Type	Description
viewiqbot	iqbot	Allows you to view the default dashboards in the IQ Bot portal.


Action	Resource Type	Description
		 Note: You must assign this permission before assigning any of the permissions below.
view	iqbotpages	Allows to view the iqbot pages licensed
utilize	iqbotpages	Allows to utilize the iqbot pages licensed

Learning instance permissions


Action	Resource Type	Description
viewlearninginstance	viewiqbot	<p>Allows you to view their learning instances in the IQ Bot portal.</p>  Note: You must assign this permission before assigning any other learning instance permission.
viewlearninginstancefromsameroles	viewlearninginstance	Allows you to view learning instances created by other you with the same role in the IQ Bot portal.
assignroletolearninginstances	viewlearninginstance	Allows you to assign some custom roles to a learning instance.

Action	Resource Type	Description
viewalllearninginstances	viewlearninginstance	Allows you to view all learning instances in the IQ Bot portal.
launchvalidation	viewlearninginstance	Allows you to access the IQ Bot Validator to review and update documents with exceptions.
createlearninginstances	viewlearninginstance	Allows you to create learning instances in the IQ Bot portal.
editlearninginstances	viewlearninginstance	Allows you to edit learning instances in the IQ Bot portal.
deletelearninginstances	viewlearninginstance	Allows you to delete their learning instances in the IQ Bot portal.
sendlearninginstancestoprod	viewlearninginstance	Allows you to send their learning instances to production in the IQ Bot portal.
trainlearninginstancegroups	viewlearninginstance	Allows you to train their learning instance groups in the IQ Bot portal.

Domains permissions

Action	Resource Type	Description
viewdomain	viewiqbot	<p>Allows you to view all domains in the IQ Bot portal.</p> <div>  Note: You must assign this permission before assigning any other domain permissions. </div>
createdomains	viewdomain	Allows you to create domains in the IQ Bot portal.
editdomains	viewdomain	Allows you to edit domains in the IQ Bot portal.
importdomains	viewdomain	Allows you to import domains in the IQ Bot portal.
exportdomains	viewdomain	Allows you to export domains in the IQ Bot portal.
deletedomains	viewdomain	Allows you to delete domains in the IQ Bot portal.

Administration permissions

Action	Resource Type	Description
viewadministration	viewiqbot	<p>Allows you to access the Administration tab in the IQ Bot portal.</p> <div>  Note: You must assign this permission before assigning any other administration permissions. </div>

Action	Resource Type	Description
viewandmanagesettings	viewadministration	Allows you to manage the IQ Bot portal advanced configuration settings.
viewandmanagemigration	viewadministration	Allows you to access the migration utility to import and export learning instances in the IQ Bot portal.

Parent topic: [Roles and permissions](#)


MetaBot permission

MetaBots are obsolete in Automation 360. However, this permission is valid to ensure that all functionality previously available is supported in Automation 360.

Access to MetaBot Designer

Permission/feature ID is not available.

Bot Creator users can access MetaBot Designer to view, create, and update MetaBots.

 **Note:** This feature is available for internal use only.

Parent topic: [Roles and permissions](#)

Package manager permissions

Enable users to view and manage packages.

Action	Resource Type	Description
view	packagemanager	Allows you to view packages.
manage	packagemanager	Allows you to view and manage packages.

Parent topic: [Roles and permissions](#)

Workload permissions

Enable you to create and manage workitems, workitem models, queues, and automations in the Control Room.

Action	Resource Type	Description
myschedule	taskscheduling	Allows you to view and manage their own schedules.
view	queue	Allows you to view their own queues.
create	queue	Allows you to create and manage their own queues.
calculate	sla	Allows you to calculate workload Service Level Agreements (SLA).
view	workload	Legacy, not used and will be removed from the future releases.
accessresourceany	queue	Legacy, not used and will be removed from the future releases.

Note: A user with the `AAE_Queue Admin` role has all the above permissions. In addition, the `AAE_Queue Admin` is able to manage all the queues in the Control Room. See [System-created roles](#).

Parent topic: [Roles and permissions](#)

Audit API

Use the Audit API to request audit data for a given input combination of date filter, sorting mechanism, and pagination.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- Users with the **AAE_Admin** role or users with the **View everyone audit log actions** permission can view audit logs for the Control Room.

Procedure

1. All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
2. Apply filters to perform basic conditional queries and pagination control for processing web pages. There are three basic features related to filtering: filtering conditions, sorting columns, and pagination parameters.

[Filtering, pagination, and sorting](#)

3. Use the POST method and endpoint URL: `<your_control_room_url>/v1/audit/messages/list`.

The following example requests unsuccessful login attempts for the month of December, 2019.

Request body:

```
{
  "sort": [
    {
      "field": "createdOn",
      "direction": "desc"
    }
  ],
  "filter": {
    "operator": "and",
    "operands": [
      {
        "operator": "gt",
        "field": "createdOn",
        "value": "2019-12-01T00:00:00.001Z"
      }
    ]
  }
}
```

```

    },
    {
      "operator": "lt",
      "field": "createdOn",
      "value": "2019-12-31T23:59:59.999Z"
    },
    {
      "operator": "eq",
      "field": "status",
      "value": "Unsuccessful"
    },
    {
      "operator": "substring",
      "field": "activityType",
      "value": "LOGIN"
    },
    {
      "operator": "substring",
      "field": "userName",
      "value": "joe.typical@myemiil.com"
    }
  ]
},
"page": {
  "length": "1000",
  "offset": "0"
}
}

```

4. Send the request.

The response for this example returns data for the date filter, sorting, and pagination. If no filtering is used in the request, a successful response returns all pages for the specified Control Room.

Response body:

```

{
  "page": {
    "offset": 0,

```

```
"total": 731064850,
"totalFilter": 9
},
"list": [
  {
    "id": "XlHj6G4BFXSp00ji5B7S",
    "eventDescription": "User does not exist in Control Room.",
    "activityType": "LOGIN",
    "environmentName": "",
    "hostName": "12.xxx.xx.x",
    "userName": "joe.typical@myemiil.com",
    "status": "Unsuccessful",
    "source": "Control Room",
    "objectName": "N/A",
    "detail": "",
    "createdOn": "2019-12-09T04:21:19Z",
    "requestId": "04965c2e-82e0-4ce4-a88d-bebe1dc3a2a8",
    "createdBy": "0"
  },
  {
    "id": "g1Hj6G4BFXSp00ji2Rwx",
    "eventDescription": "User does not exist in Control Room.",
    "activityType": "LOGIN",
    "environmentName": "",
    "hostName": "12.xxx.xx.x",
    "userName": "joe.typical@myemiil.com",
    "status": "Unsuccessful",
    "source": "Control Room",
    "objectName": "N/A",
    "detail": "",
    "createdOn": "2019-12-09T04:21:16Z",
    "requestId": "61672553-477d-4012-ab47-2a27f6553c4e",
    "createdBy": "0"
  },
  .....
  {
```

```
{
  "id": "ETyk6G4BFXSp00jiaJjt",
  "eventDescription": "User does not exist in Control Room.",
  "activityType": "LOGIN",
  "environmentName": "",
  "hostName": "12.xxx.xx.x",
  "userName": "joe.typical@myemiil.com",
  "status": "Unsuccessful",
  "source": "Control Room",
  "objectName": "N/A",
  "detail": "",
  "createdOn": "2019-12-09T03:11:58Z",
  "requestId": "eb01de-1f81-4a7c-8978-405806e146bd",
  "createdBy": "0"
}
```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Example of createdOn date and userName filters in Audit API

Create a filter that retrieves audit log entries for a specified date range for a user with a specific value in the **userName** field.

Use filtering to help narrow your results. The following example identifies unsuccessful login attempts for users with the value **"john, doe"** in the **userName** field from December 1, 2020 through December 31, 2020.

Request body:

```
{
  "sort": [
    {
      "field": "createdOn",
      "direction": "desc"
    }
  ],
  "filter": {
    "operator": "and",
```

```

"operands": [
  {
    "operator": "gt",
    "field": "createdOn",
    "value": "2020-12-01T00:00:00.001Z"
  },
  {
    "operator": "lt",
    "field": "createdOn",
    "value": "2020-12-31T23:59:59.999Z"
  },
  {
    "operator": "eq",
    "field": "status",
    "value": "Unsuccessful"
  },
  {
    "operator": "substring",
    "field": "activityType",
    "value": "LOGIN"
  },
  {
    "operator": "substring",
    "field": "userName",
    "value": "john,doe"
  }
]
},
"page": {
  "length": "1000",
  "offset": "0"
}
}

```

This request identified three audit log entries out of 731,148.339 entries from this Control Room log entries.

Response body:

```
{
  "page": {
    "offset": 0,
    "total": 731148339,
    "totalFilter": 3
  },
  "list": [
    {
      "id": "kLjB8G4BFXSp0OjioMK1",
      "eventDescription": "User does not exist in Control Room.",
      "activityType": "LOGIN",
      "environmentName": "",
      "hostName": "50.xxx.xxx.xx",
      "userName": "john,doe@mycompany.com",
      "status": "Unsuccessful",
      "source": "Control Room",
      "objectName": "N/A",
      "detail": "",
      "createdOn": "2020-12-10T17:00:52Z",
      "requestId": "3c0f8e47-5820-43e8-b2b3-83b2f1cb86c9",
      "createdBy": "0"
    },
    {
      "id": "SLjB8G4BFXSp0Ojikl5i",
      "eventDescription": "User does not exist in Control Room.",
      "activityType": "LOGIN",
      "environmentName": "",
      "hostName": "50.xxx.xxx.xx",
      "userName": "john,doe@mycompany.com",
      "status": "Unsuccessful",
      "source": "Control Room",
      "objectName": "N/A",
      "detail": "",
      "createdOn": "2020-12-10T17:00:48Z",
      "requestId": "eba3e5a7-0034-440a-a786-110a84fea7c9",
      "createdBy": "0"
    }
  ]
}
```

```

    },
    {
      "id": "7bjB8G4BFXSp00jicEGO",
      "eventDescription": "User does not exist in Control Room.",
      "activityType": "LOGIN",
      "environmentName": "",
      "hostName": "50.xxx.xxx.xx",
      "userName": "john,doe",
      "status": "Unsuccessful",
      "source": "Control Room",
      "objectName": "N/A",
      "detail": "",
      "createdOn": "2020-12-10T17:00:39Z",
      "requestId": "64184450-aad5-4024-bcf5-491fb5276d0c",
      "createdBy": "0"
    }
  ]
}

```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Device API

Identify all available users with unattended Bot Runner licenses, or filter for users by name.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

Return available Bot Runners

Return a list of available users with unattended Bot Runner licenses. This endpoint returns the user id, which is a numeric value that is used by APIs to identify users.

```
POST <control_room_URL>/v1/devices/runasusers/list
```


List available unattended Bot Runners API

Return a list of available users with unattended Bot Runner licenses. This endpoint returns the user id, which is a numeric value that is used by APIs to identify users.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- You must be assigned a custom role that is associated with a Run As user device.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST method and endpoint URL: `<your_control_room_url>/v1/devices/runasusers/list`.

Request body:

```
{
  "sort": [
    {
      "field": "username",
      "direction": "asc"
    }
  ],
  "filter": {},
  "page": {}
}
```

[Filtering, pagination, and sorting](#)

3. Send the request.

Response body: In a successful request, this endpoint returns the following data:

- `id`: a unique numeric identifier for a user with the Bot Runner license.

- **device** : if the user is configured with a default device, the device name is returned. For example, **DESKTOP-DBO6SIE** . Otherwise, this parameter returns the message **Picked at run time** , indicating that a device must be selected from a device pool in order to run a bot.

```
{
  "page":{
    "offset":0,
    "total":6,
    "totalFilter":6
  },
  "list":[
    {
      "id":"9",
      "username":"ubr01_rt",
      "device":"DESKTOP-DBO6SIE",
      "deviceId":"3"
    },
    {
      "id":"10",
      "username":"ubr02_rt",
      "device":"DESKTOP-DBO6SIE",
      "deviceId":"3"
    },
    {
      "id":"11",
      "username":"ubr03_rt",
      "device":"DESKTOP-DBO6SIE",
      "deviceId":"3"
    },
    {
      "id":"12",
      "username":"ubr04_rt",
      "device":"Picked at run time",
      "deviceId":"-1"
    },
    {
      "id":"13",
```

```
    "username": "ubr05_rt",
    "device": "Picked at run time",
    "deviceId": "-1"
  },
  {
    "id": "14",
    "username": "ubr06_rt",
    "device": "Picked at run time",
    "deviceId": "-1"
  }
]
```

Next steps

If you are performing the steps to deploy a bot or to create an automation schedule, and the user associated with the Bot Runner license does not have a default device assigned or if you want to select a different device, perform this task: [List device pools API](#).

To deploy a bot that runs on the default device assigned to the Bot Runner user, perform this task: [Bot deployment - V3](#).

To create an automation schedule with a bot that runs on the default device assigned to the Bot Runner user, perform this task: [Schedule bot to run API](#).

Assign default device API

Use the default device allocation API to set a specific device as the default deployment device for the specific user.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- You must have User management rights or admin rights.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST, PUT, GET, DELETE method and endpoint URL: `<your_control_room_url>/runasusers/default`.

Request body:

Enter the valid `deviceId` and `userId`.

```
{
  "deviceId": 1046,
  "userId": 589
}
```

3. Send the request.

When the request is successful, a default device is allocated to the specified user. Now this device will be used as default for the deployment.

Response body:

In this example, the `deviceId` with value 1046 is allocated as default device for the specified user with `userId` 589.

```
{
  "deviceId": 1046,
  "userId": 589
}
```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Trigger API

Map triggers to users or roles for an attended Bot Runner user by using the Trigger API. With the Trigger API, you can also create and delete event triggers.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- **Create an event trigger**

Create an event trigger for a Bot Runner user, role, or bot file. Ensure that the associated users and roles have a Bot Runner license.

- **Delete an event trigger**

Delete an event trigger that is associated with a user, role, or bot.

Parent topic: [Control Room APIs](#)

Create an event trigger

Create an event trigger for a Bot Runner user, role, or bot file. Ensure that the associated users and roles have a Bot Runner license.

Request

```
POST http://{{ControlRoomURL}}/v1/triggers/triggermapping
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request body for users:

```
{
  "botFileId":106587,
  "botFileLabel":"string",
  "users":{
    "userIds":[
      "985"
    ]
  }
}
```

Request body for roles:



```
{
  "botFileId":106587,
  "botFileLabel":"string",
  "roles":{
    "roleIds":[
      721, 645
    ]
  }
}
```

```

    ]
  }
}

```

Request Parameters

Parameter	Type	Required	Description
botFileId	Integer	Yes	Unique identifier of the bot file.
botFileLabel	String	No	Bot file label. It can be PRODUCTION or empty for the latest version.
userIds	Array	Yes*	<p>Enter the IDs of the user. Only the users listed here will be associated with the trigger.</p> <div> Note: The user associated with these <code>userIds</code> must have a Bot Runner license to run the API.</div>
roleIds	Array		<p>Enter the IDs of the role. Only the roles listed here will be associated with the trigger.</p> <div> Note: The users associated with these <code>roleIds</code> must have a Bot Runner license to run the API.</div>

**One of the previously mentioned parameters is required to create an event trigger.*

Response

```

{
  "triggerMappings": [
    {

```

```

    "id": "399",
    "userId": "985",
    "botPath": "Automation Anywhere\\Bots\\botA",
    "modifiedBy": "289",
    "lastModified": "2022-04-06T09:09:43.893256Z",
    "botName": "botA",
    "botId": "106587",
    "botLabel": "string"
  }
]
}

```

Response Parameters

Parameter	Type	Description
triggerMappings	Array	<p>The triggerMappings array returns with all the details of the triggerMapping created. Each array element contains the following values:</p> <ul style="list-style-type: none"> • id: ID of the trigger mapping • userId: ID of the user • roleId: ID of the role • botPath: The path of the bot • modifiedBy: The ID of the user that did the triggerMapping • lastModified: The timestamp of the triggerMapping • botName: The name of the bot that got the triggerMapping. • botId - The Id of the bot that got the triggerMapping • botLabel: The label of the bot that got the triggerMapping • licenseFeatures: License assigned to the user

Parent topic: [Trigger API](#)

Delete an event trigger

Delete an event trigger that is associated with a user, role, or bot.

Request

```
DELETE http://{{ControlRoomURL}}/v1/triggers/triggermapping/{id}
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request Parameters

Parameter	Type	Description
Id	Integer	Event trigger identifier that you want to delete

Response


```
204 No Content
The relationship was deleted.
```

For more information about return codes, see [API response codes](#).

Parent topic: [Trigger API](#)

Credential Vault APIs

As an Control Room user with **Manage my credentials and lockers** feature permissions, you have the option to use the Credential Vault API to manage your attributes, credentials, and lockers in the Control Room.

 **Note:** You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

By default, all users can create credentials. You are the Credential owner of any credentials that you created. As a Credential owner, you can update, delete, and transfer the ownership of your credentials.

Configure the Credential Vault for your organization

As a Locker admin, configure a locker and a credential, then assign the credential to the locker. It is also acceptable to first configure a credential, then a locker.

1. [Authenticate the user.](#)

Use the POST method to generate an authentication JSON Web Token.

2. [Configure a locker using API](#)
3. [Configure a credential with attribute values using API](#)
4. [Assign credential to locker API](#)

Manage your credentials

Authenticate yourself as a basic user to retrieve the list of credentials that you have access to then assign a value to a specific attribute.

1. [Authenticate the user.](#)

Use the POST method to generate an authentication JSON Web Token.

2. [List credentials using API](#)
3. [Update attribute values](#)

Retrieve masked credentials

You will be able to retrieve your masked credentials, if you have **View and edit ALL credentials attributes value** permissions.

1. [Authenticate the user.](#)

Use the POST method to generate an authentication JSON Web Token.

2. Get the Masked Attribute Value (For more information, see [Get Masked credentials](#))
 - a. Get Credential ID.
 - b. Get Credential Attribute ID.
 - c. Use the Credential ID and Credential Attributed ID to retrieve the masked attribute value.

- **Set device login credentials API**

Use the login setting endpoint of the Credential Vault API to update the user name and password for a device. You can use this endpoint to set or update the login credentials for your own device without additional permissions.

- **Configure a locker using API**

Use a combination of endpoints to create a locker and assign locker access permissions to users.

- **[Configure a credential with attribute values using API](#)**
Create a credential with a standard attribute and add two additional attributes with user-input values.
- **[Assign credential to locker API](#)**
Add a credential to a locker to enable other users to access the credential to build and run bots.
- **[List credentials using API](#)**
Return a list of the credentials for which you are the owner or have access through a locker. If you have the **AAE_Locker Admin** role, this endpoint returns all the credentials in the Control Room.
- **[Update attribute values](#)**
Update either a standard or user-input value to an attribute, based on your access permissions to the credential. A standard value is accessible by all users of the credential; an attribute with a user-input value enables each user to provide their own value which the other users cannot access.
- **[Get Masked credentials](#)**
Use the `attribute values` endpoint of the Credential Vault API to get the masked credentials.

Parent topic: [Control Room APIs](#)

Set device login credentials API

Use the login setting endpoint of the Credential Vault API to update the user name and password for a device. You can use this endpoint to set or update the login credentials for your own device without additional permissions.

Request

To set or update the login credentials on other users' devices, for example, to deploy bots on unattended Bot Runners, you must have the AAE_admin role or a custom role with the `Bot Auto-Login Credentials API` permission. To update the credentials for your own device, you only need the device username and password:

```
PUT http://{{ControlRoomURL}}/v2/credentialvault/loginsetting
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request body:

```
{
  "loginUsername": "aai\\jane.doe",
```

```
"loginPassword":"Automation123"
}
```

To update the credentials for another user's device, you must also include either the username or userid of that user, as demonstrated in this code example:

```
{
  "loginUsername":"aai\\jane.doe",
  "loginPassword":"Automation123",
  "username":"john-doe"
}
```

Request Parameters

Parameter	Type	Description
loginUsername	String	Enter the device login user name.
loginPassword	String	Enter the device login password.
username	String	Enter the user name of the user's device you want to update/set.

Response

```
"Credentials updated for jane"
```

Note: For an Control Room that is deployed on Cloud and has SAML authentication enabled, generate the web token with your `username` and `apikey`.

[Authenticate \(username and apiKey\)](#)

Parent topic: [Credential Vault APIs](#)

Configure a locker using API

Use a combination of endpoints to create a locker and assign locker access permissions to users.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- You must be assigned the **AAE_Admin**, **AAE_Locker Admin** role or have a custom role that includes the **Manage my lockers** permission.

You will provide the role ID to assign consumer access to users [List roles](#).

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST method and endpoint URL: `<your_control_room_url>/v2/credentialvault/lockers` to create the locker.

Request body:

```
{
  "name": "HumanResourcesCredentials",
  "description": "Login credentials for the HR dept"
}
```

3. Send the request.

Response body: In a successful request, this endpoint returns the `id`, which is a unique numeric identifier for the locker. Use the locker ID in subsequent API requests, such as to add consumers or credentials to the locker.

```
{
  "id": "1551",
  "name": "HumanResourcesCredentials",
  "description": "Login credentials for the HR dept",
  "createdBy": "1508",
  "createdOn": "2020-12-28T22:24:40.462253Z",
  "updatedBy": "1508",
  "updatedOn": "2020-12-28T22:24:40.462259Z",
}
```

```
"version": "0"
}
```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Assign locker access permissions to users. [Locker permissions](#)

4. **Optional:** Assign another locker owner. Use the PUT method and endpoint

URL: `<your_control_room_url>/v2/credentialvault/lockers/{lockerId}/members/{userId}`.

 **Note:** The locker creator is automatically assigned the locker owner permission.

Request body:

```
{
  "permissions": [
    "own"
  ]
}
```

5. Send the request.

Response body: This endpoint does not return data.

6. **Optional:** Assign a locker manager. Use the PUT method and endpoint

URL: `<your_control_room_url>/v2/credentialvault/lockers/{lockerId}/members/{userId}`.

Request body:

```
{
  "permissions": [
    "manage"
  ]
}
```

7. Send the request.

Response body: This endpoint does not return data.

8. **Optional:** Assign a locker participant. Use the PUT method and endpoint

URL: `<your_control_room_url>/v2/credentialvault/lockers/{lockerId}/members/{userId}`.

Request body:

```
{
  "permissions": [
    "participate"
  ]
}
```

9. Send the request.

Response body: This endpoint does not return data.

10. Assign locker consumers. Use the POST method and endpoint URL::

`<your_control_room_url>/v2/credentialvault/lockers/{lockerId}/consumers`

Request body: Provide the role ID. All users who are assigned that custom role can build and run bots using the credentials in this locker, as well as enter values into credentials that accept user-provided attribute values.

```
{
  "id": "516"
}
```

11. Send the request.

Response body: This endpoint does not return data.

Next steps

If you are following the steps to configure your Credential Vault, do this next: [Assign credential to locker API](#).

Parent topic: [Credential Vault APIs](#)

Configure a credential with attribute values using API

Create a credential with a standard attribute and add two additional attributes with user-input values.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

- All roles have the **Manage my credentials and lockers** permission that is necessary to configure credentials and attributes. No additional permissions are necessary to use this endpoint.

In this example, you configure a credential with three attributes to hold email hostname, username, and password.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.

Create a credential:

2. Use the POST method and endpoint URL: **<your_control_room_url>/v2/credentialvault/credentials**.

Request body: This example request includes the following required parameters:

- **userProvided**: a boolean value that configures whether the attribute requires a user's input (**true**) or is standard for all users (**false**).
- **masked**: a boolean value that configures whether the attribute value is masked with asterisks (**true**) or is visible to users (**false**).

```
{
  "name": "Email",
  "attributes": [
    {
      "name": "hostname",
      "userProvided": false,
      "masked": false
    }
  ]
}
```

3. Send the request.

Response body: In a successful request, this endpoint returns the following data:

- **id**: a unique numeric identifier for the credential.
- **attributes:id**: a unique numeric identifier for attribute.

```
{
  "id": "1630",
  "name": "Email",
  "description": "",
```

```
"ownerId": "1508",
"attributes": [
  {
    "id": "3335",
    "name": "hostname",
    "description": "",
    "userProvided": false,
    "masked": false,
    "createdBy": "1508",
    "createdOn": "2020-12-28T22:04:41.366448Z",
    "updatedBy": "1508",
    "updatedOn": "2020-12-28T22:04:41.366450Z",
    "version": "0",
    "passwordFlag": false
  }
],
"createdBy": "1508",
"createdOn": "2020-12-28T22:04:41.366460Z",
"updatedBy": "1508",
"updatedOn": "2020-12-28T22:04:41.366464Z",
"version": "0"
}
```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Assign a standard value to the hostname attribute to the credential:

4. Use the POST method and endpoint URL: `<your_control_room_url>/v2/credentialvault/credentials/{credentialId}/attributevalues`.

```
{
  "list": [
    {
      "credentialAttributeId": "3335",
      "value": "mail.example.com"
    }
  ]
}
```



```
]
}
```

5. Send the request.

Response body:

```
{
  "list": [
    {
      "id": "1630",
      "credentialAttributeId": "3335",
      "value": "mail.example.com",
      "userId": "1508",
      "createdBy": "1508",
      "createdOn": "2020-12-28T22:04:41.366460Z",
      "updatedBy": "1508",
      "updatedOn": "2020-12-28T22:04:41.366464Z",
      "version": "0"
    }
  ]
}
```

6. Add the `username` and `password` attributes to the `Email` credential. Use the PUT method and endpoint URL: `<your_control_room_url>/v2/credentialvault/credentials/{credentialId}`.

Note: You must include the existing attributes along with the new attributes in the request, otherwise the current attributes will be overwritten.

Request body: Since you have specified the credential ID in the request URL, it is not required to include the credential ID or name in the request body. In this example request body, the `username` and `password` attributes are configured with values that accept a different input from each user. Additionally, the password attribute is configured to mask the entered value with asterisks.

```
{
  "attributes": [
    {
      "name": "username",
```

```
    "userProvided": true,  
    "masked": false  
  },  
  {  
    "name": "password",  
    "userProvided": true,  
    "masked": true  
  },  
  {  
    "name": "hostname",  
    "userProvided": false,  
    "masked": false  
  }  
]  
}
```

7. Send the request.

Response body: The response body returns the credential with details of the three attributes.

```
{  
  "id": "1630",  
  "name": "Email",  
  "description": "",  
  "ownerId": "1508",  
  "attributes": [  
    {  
      "id": "3335",  
      "name": "hostname",  
      "description": "",  
      "userProvided": false,  
      "masked": false,  
      "createdBy": "1508",  
      "createdOn": "2020-12-28T22:04:41.366448Z",  
      "updatedBy": "1508",  
      "updatedOn": "2020-12-28T22:04:41.366450Z",  
      "version": "0",  
      "passwordFlag": false  
    }  
  ]  
}
```

```
    },
    {
      "id": "3336",
      "name": "username",
      "description": "",
      "userProvided": true,
      "masked": false,
      "createdBy": "1508",
      "createdOn": "2020-12-28T22:04:41.366450Z",
      "updatedBy": "1508",
      "updatedOn": "2020-12-28T22:04:41.366450Z",
      "version": "0",
      "passwordFlag": false
    },
    {
      "id": "3337",
      "name": "password",
      "description": "",
      "userProvided": true,
      "masked": true,
      "createdBy": "1508",
      "createdOn": "2020-12-28T22:04:41.366450Z",
      "updatedBy": "1508",
      "updatedOn": "2020-12-28T22:04:41.366450Z",
      "version": "0",
      "passwordFlag": false
    }
  ],
  "createdBy": "1508",
  "createdOn": "2020-12-28T22:04:41.366460Z",
  "updatedBy": "1508",
  "updatedOn": "2020-12-28T22:06:35.366464Z",
  "version": "2"
}
```

Next steps

If you are following the steps to configure your Credential Vault, do this next: [Assign credential to locker API](#).

Parent topic: [Credential Vault APIs](#)

Assign credential to locker API

Add a credential to a locker to enable other users to access the credential to build and run bots.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- You must have **own**, **manage**, or **participate** access permissions to the locker.

To assign a credential to a locker, you provide the credential and locker IDs in the URL.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the PUT method and endpoint URL:: `<your_control_room_url>/v2/credentialvault/lockers/{lockerId}/credentials/{credentialId}`.

Request body: This endpoint does not have a request body.

3. Send the request.

Response body: This endpoint does not return data.

Parent topic: [Credential Vault APIs](#)

List credentials using API

Return a list of the credentials for which you are the owner or have access through a locker. If you have the **AAE_Locker Admin** role, this endpoint returns all the credentials in the Control Room.

Request

```
POST http://{{ControlRoomURL}}/v2/credentialvault/credentials/list
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

You must have access to the credential, either as the credential owner or through a locker.

Request body:

```
{
  "filter": {
    "operator": "and",
    "operands": [
      {
        "operator": "gt",
        "field": "createdOn",
        "value": "2022-10-12T00:00:00.123Z"
      },
      {
        "operator": "lt",
        "field": "createdOn",
        "value": "2022-10-12T23:00:00.123Z"
      }
    ]
  },
  "sort": [
    {
      "field": "name",
      "direction": "asc"
    }
  ],
  "page": {
    "offset": 0,
    "length": 100
  }
}
```

Request Parameters

Parameter	Type	Required	Description
consumed	Boolean	No	Filter credentials by fact if credential is user Provided and consumed by current user.
sort	Array	No	<p>By default, search results are sorted in ascending order of the IDs. To specify an alternate sorting, use the sort query parameter.</p> <p>Enter the field by which you want to sort along with the direction asc (ascending) or desc (descending).</p>
filter	Object	No	<p>Filters the result based on operator, field, or value.</p> <p>operator Allowed enumerations are NONE, lt, le, eq, ne, ge, gt, substring, and, or, not.</p> <p>field Allowed values are name, lastModified, path, or folder.</p> <p>value Specify a value for the name, lastModified, path, or folder that you have selected in the field parameter.</p>
page	Object	No	The page object allows you to get the desired number of pages.

For more information on Filtering, Pagination, and Sorting, see [Filtering, pagination, and sorting](#).

Response

```
{
  "page": {
    "offset": 0,
```

```
"total": 7,  
  "totalFilter": 1  
},  
"list": [  
  {  
    "id": "307",  
    "name": "Sample-Credential",  
    "description": "Test credential Created from API request",  
    "completed": true,  
    "lockerId": "15",  
    "ownerId": "21",  
    "attributes": [  
      {  
        "id": "916",  
        "name": "Username",  
        "description": "Username for a sample API call",  
        "userProvided": false,  
        "masked": false,  
        "passwordFlag": false,  
        "createdBy": "21",  
        "createdOn": "2022-10-28T00:15:09.319987Z",  
        "updatedBy": "21",  
        "updatedOn": "2022-10-28T00:15:09.319988Z",  
        "version": "0"  
      }  
    ],  
    "createdBy": "21",  
    "createdOn": "2022-10-12T20:42:32.896315Z",  
    "updatedBy": "21",  
    "updatedOn": "2022-10-12T20:42:32.896317Z",  
    "version": "0"  
  }  
]  
}  
400  
Bad Request
```

Response parameters

Parameter	Type	Description
offset	Integer	The starting list offset, used for pagination.
total	Integer	Total number of records.
totalFilter	Integer	Number of records after applying the filter.
List	Array	The list of directories and files.
List objects		
id	Integer	The unique ID of the credential.
name	String	Name of the credential.
description	String	Description of credential
completed	Boolean	Shows if consumer(s) needs to provide a value for user-provided attribute or not
lockerId	String	Id of the locker which has this credential, can be null if the credential is not assigned to any locker
ownerId	String	Id of Credential owner User
attributes	Object	<p>Displays the credential attributes:</p> <pre> id Id of credential attribute name Name of credential description Description of credential attribute </pre>

Parameter	Type	Description
		<p>userProvided A flag to indicate if the credential attribute is user-provided or common</p> <p>masked A flag to indicate if the credential attribute value is masked</p> <p>passwordFlag A flag to indicate if the credential attribute used for password is a masked input fields</p> <p>createdBy Id of the user who created the object</p> <p>createdOn Creation date of the object</p> <p>updatedBy Id of the user who made a latest update on the object</p> <p>updatedOn Latest updated date of the object</p> <p>version Version of the object, has to be provided by the client to track simultaneous updates</p>
createdBy	String	Id of the user who created the object
createdOn	String	Creation date of the object.
updatedBy	String	Id of the user who made a latest update on the object
updatedOn	String	Latest updated date of the object
version	String	Version of the object, has to be provided by the client to track simultaneous updates

Now that you have the IDs of attributes that accept a user-input value, next [Update attribute values](#).

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Parent topic: [Credential Vault APIs](#)

Update attribute values

Update either a standard or user-input value to an attribute, based on your access permissions to the credential. A standard value is accessible by all users of the credential; an attribute with a user-input value enables each user to provide their own value which the other users cannot access.

Request

```
POST https://{ControlRoomURL}/v2/credentialvault/credentials/{credentialId}/attributevalues
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Tip:

- To set the standard value, you must have access to the credential, either as the credential owner or as a locker Admin or Manager.
- To set the user-input value, you must have access to the credential, either as the credential owner or as a locker consumer.
- Verify whether the attribute accepts a standard or user-input value. This is indicated in the userProvided output parameter. [List credentials using API](#).

Request body:

```
{
  "list": [
    {
      "credentialAttributeId": "890",
      "value": "aVerySecurePassword"
    }
  ]
}
```

```

    ]
  }

```

Request Parameters

Parameter	Type	Required	Description
credentialAttributeId	String	Yes	Id of credential attribute.
value	String	Yes	Value of credential attribute.

Response

```
201 Created
```

For more information on the return codes, see [API response codes](#).

```

{
  "list": [
    {
      "id": "131",
      "credentialAttributeId": "890",
      "value": "00172qLH9JgAHUB21vCGj4ZheVokDL6unV1HIX8rWUw=",
      "userId": "38",
      "createdBy": "38",
      "createdOn": "2022-09-22T14:42:02.876540Z",
      "updatedBy": "38",
      "updatedOn": "2022-09-22T14:42:02.876544Z",
      "version": "0",
      "password": false
    }
  ]
}

```

Response Parameters

Parameter	Type	Description
id	String	Id of credential attribute value.
credentialAttributeld	String	Id of credential attribute.
value	String	Value of credential attribute.
userId	String	Id of user to which the value belongs to.
createdBy	String	Id of the user who created the Credential.
createdOn	String	Date of creation of the Credential.
updatedBy	String	Id of the user who made a latest update on the Credential.
updatedOn	String	Date of the latest update of the Credential.
version	String	Version of the Credential, has to be provided by the client to track simultaneous updates.
password	Boolean	Flag to indicate if there is a password set to the Credential.

 **Note:** You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Parent topic: [Credential Vault APIs](#)

Get Masked credentials

Use the `attribute values` endpoint of the Credential Vault API to get the masked credentials.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- You will be able to retrieve your masked attributes using the API and with **View and edit ALL credentials attributes** permission.

1. Use the POST method and endpoint URL: `<control_room_URL>/v2/credentialvault/credentials/list`. **Request body:**

The following example will look for the credential called *ED10355* using POST with the endpoint

```
POST https://{ControlRoomURL}/v2/credentialvault/credentials/list
```

```
{
  "filter":{
    "operator":"and",
    "operands":[
      {
        "operator":"eq",
        "field":"name",
        "value":"ED10355"
      }
    ]
  }
}
```

2. Send the request.

Response body:

```
{
  "page": {
    "offset": 0,
    "total": 6,
    "totalFilter": 1
  },
  "list": [
```

```

{
  "id": "25",
  "name": "ED10355",
  "description": "",
  "ownerId": "132",
  "attributes": [
    {
      "id": "90",
      "name": "name",
      "description": "",
      "userProvided": false,
      "masked": true,
      "createdBy": "132",
      "createdOn": "2022-01-28T19:21:59.388237Z",
      "updatedBy": "132",
      "updatedOn": "2022-01-28T19:45:57.351698900Z",
      "version": "2",
      "passwordFlag": false
    }
  ],
  "createdBy": "132",
  "createdOn": "2022-01-28T19:21:59.388237Z",
  "updatedBy": "132",
  "updatedOn": "2022-01-28T19:45:57.353647200Z",
  "version": "2",
  "completed": false,
  "externalVaultCredentialName": ""
}
]
}

```

Note: The response shows the credential id (`"id": "25"`) and credential attribute id (`"id": "90"`). In case if you want to retrieve the attribute id, to see if it has masked values or not. Use the GET method with the `v2/credentialvault/credentials/25` to list the attributes pertaining to the credential id (`"id": "25"`).

3. To get the masked attribute value, use the GET method method with the endpoint

```
GET https://{ControlRoomURL}/v2/credentialvault/credentials/25/attributevalues?credentialAttributeId=90
```

Note: You will be able to view your masked credential attributes, only if you have the **View and edit ALL credentials attributes**. You will not be able to view the masked credential attributes of other users.

4. Send the request.

Response body:

```
{
  "list": [
    {
      "id": "46",
      "credentialAttributeId": "90",
      "value": "maskedsecret",
      "createdBy": "132",
      "createdOn": "2022-01-28T19:21:59.635310800Z",
      "updatedBy": "132",
      "updatedOn": "2022-01-28T19:40:04.495291100Z",
      "version": "1",
      "password": false
    }
  ]
}
```

Note: To get the *credentialAttributeValue* for any user-provided credential, you must provide a query parameter *UserId* must be provided in the request with the GET method and the *UserId* should belong to the user who has provided the credentials. For example:

```
GET https://{ControlRoomURL}/v2/credentialvault/credentials/25/attributevalues?credentialAttributeId=90&userId=452
```

The REST API responds to each request with an HTTP response code. For details about the response codes, see [API response codes](#).

Parent topic: [Credential Vault APIs](#)

Bot Execution Orchestrator API

As a Control Room administrator or a user with **View and Manage Scheduled Activity** permission, you can monitor the bot progress using a set of Control Room APIs.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

High-level process for monitoring bots

Searchable fields for devices:

- **hostName:** The host name of the device configured as a Bot Runner. If a naming convention is used for host names, searching on a unique **substring** in the host name is an effective way to identify Bot Runner devices.
- **userId:** The unique numeric identification for a specific user also identifies the Bot Runner device. Unique user naming conventions can be used to identify users and devices that are licensed and configured as Bot Runners.

Searchable fields for bots:

- **name:** The unique name of a bot. You can search on the exact name (**eq**) or a text string (**substring**) that is contained in the bots name.
- **path:** The relative path of a folder in the Control Room. You can search on a full path or a **substring** contained in the path.
- **[Request device details](#)**
Use this API to retrieve a list of devices that are available for bot deployment.
- **[Activity list](#)**
You can returns a list of bot executions based on filtering, ordering, and pagination rules. Use this API to fetch execution details for specific automation IDs as returned by the deployment API, with the exception of bot output variables and callback information.

Parent topic: [Control Room APIs](#)

Request device details

Use this API to retrieve a list of devices that are available for bot deployment.

Prerequisites

Roles and license

You have to authenticate as a user with an **Unattended bot runner license**.

- **URL:**

```
http://<your_control_room_url>/v2/devices/list
```

- **Method:** POST

Supported filterable parameters:

id

The numeric identifier for a device.

- **Field:** id
- **Type:** integer

```
{
  "filter": {
    "operator": "eq",
    "value": "7",
    "field": "id"
  }
}
```

hostName

The name of the registered device.

- **Field:** hostName
- **Type:** string

```
{
  "filter": {
    "operator": "substring",
    "value": "AA",
    "field": "hostName"
  }
}
```

userId

A unique numeric identifier for the user associated with the registered device.

- **Field:** userId
- **Type:** long

```
{
  "filter": {
    "operator": "eq",
    "value": "13",
    "field": "userId"
  }
}
```

status

The connection status of device.

- **Field:** status
- **Type:** string

```
{
  "filter": {
    "operator": "eq",
    "value": "CONNECTED",
    "field": "status"
  }
}
```

This task requests a list of all devices with a specific string in the hostname parameter and specific status of the device. Use the list in the response to identify which devices are connected and available to run bots.

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Procedure

1. Select the **POST** method.
2. Enter the URL for the API:

```
https://<your_control_room_url>/v2/devices/list
```

3. In the request body, add the filtering, sorting, and pagination rules to retrieve the device list that you want to deploy.

Note: The `fields` array filter parameter in the request body is currently not supported. When you send the field name in the request body to restrict the number of fields in the response, it does not work as expected and instead returns all the fields.

For example, this request body uses "and" as `operator` and the device "status" and "hostname" as `field` to filter the required results. The results will be sorted in "descending" order based on "status".

```
{
  "sort": [
    {
      "field": "status",
      "direction": "desc"
    }
  ],
  "filter": {
    "operator": "and",
    "operands": [
      {
        "operator": "eq",
        "value": "CONNECTED",
        "field": "status"
      },
      {
        "operator": "substring",
        "value": "win",
        "field": "hostName"
      }
    ]
  },
  "page": {
    "offset": 0,
```

```

    "total": 71,
    "totalFilter": 18,
    "length": 100
  }
}

```

4. Send the request.

- In a REST client, click **SEND**.
- In the Swagger interface, click **Execute**.

Response body:

The response returns the details of two devices that are in the "connected" **status** and for which the **hostName** starts with "win" based on the requested filter criteria.

```

{
  "page": {
    "offset": 0,
    "total": 71,
    "totalFilter": 7
  },
  "list": [{
    "id": "163",
    "type": "ATTENDED_BOT_RUNNER",
    "hostName": "winwlm-2",
    "userId": "",
    "userName": "",
    "status": "CONNECTED",
    "poolName": "",
    "fullyQualifiedHostName": "-",
    "updatedBy": "b2",
    "updatedOn": "2020-07-07T08:24:56.091061Z",
    "botAgentVersion": "12.1"
  }, {
    "id": "162",
    "type": "ATTENDED_BOT_RUNNER",
    "hostName": "winwlm-1",
    "userId": "",

```

```

    "userName": "",
    "status": "CONNECTED",
    "poolName": "",
    "fullyQualifiedHostName": "-",
    "updatedBy": "b1",
    "updatedOn": "2020-07-07T08:24:55.982047Z",
    "botAgentVersion": "12.1"
  }
}

```

Next steps

You can use the device IDs received in the response to deploy the bots on Bot Runners.

Parent topic: [Bot Execution Orchestrator API](#)

Activity list

You can return a list of bot executions based on filtering, ordering, and pagination rules. Use this API to fetch execution details for specific automation IDs as returned by the deployment API, with the exception of bot output variables and callback information.

Request

```
POST https://{ControlRoomURL}/v3/activity/list
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request body:

```

{
  "sort": [
    {
      "field": "endTime",
      "direction": "desc"
    }
  ],

```

```

    "filter":{
      "operator":"eq",
      "value":"UPDATE",
      "field":"status"
    },
    "page":{
      "length":100,
      "offset":0
    }
  }
}

```

Filter the results using `automationId`, `deviceId`, `status`, or `deploymentId`. You can also use a combination of these filters to optimize your search results.

Request body using deployment ID:

```

{
  "filter": {
    "operator": "eq",
    "field": "deploymentId",
    "value": "14c2b6f8-c2a0-4a57-959d-ef413df0d179"
  }
}

```

Note: Use the empty filter or no filter to retrieve all the information in the search results.

Request parameters

Parameter	Type	Required	Description
sort	Array	No	<p>By default, search results are sorted in descending order with respect to their IDs. An alternative sorting is specified using the sort query parameter.</p> <p>Enter the field by which you want to sort along with the direction <code>asc</code> (ascending) or <code>desc</code></p>

Parameter	Type	Required	Description
			(descending). For more information on sorting, see Filtering, pagination, and sorting .
filter	Object	No	Filters the result. For more information on sorting, see Filtering, pagination, and sorting .
fields	Array	No	Filter the result based on the fields.
page	Object	No	The page object allows you to get the desired pages. For more information on pagination rules, see Filtering, pagination, and sorting .

Response

200 OK

For more information on return codes, see [API response codes](#).

```
{
  "page": {
    "offset": 0,
    "total": 2387,
    "totalFilter": 2
  },
  "list": [
    {
      "id": "44266e73-4688-4c5a-bad4-11f4c734804d_653a835e07a597fc",
      "automationName": "7 Minutes Bot.22.04.22.13.01.23.chintan-runner"
    },
    {
      "fileName": "7 Minutes Bot",
      "filePath": "",
      "type": "RUN_NOW",
      "startDateTime": "2022-04-22T07:31:44.527762800Z",
      "endDateTime": "1970-01-01T00:00:00Z",
    }
  ]
}
```

```

    "command": "",
    "status": "UPDATE",
    "progress": 0,
    "automationId": "",
    "userId": "1141",
    "deviceId": "352",
    "currentLine": 5,
    "totalLines": 0,
    "fileId": "139836",
    "modifiedBy": "1140",
    "createdBy": "1140",
    "modifiedOn": "2022-04-22T07:31:45.265728Z",
    "createdOn": "",
    "deploymentId": "b969e264-65bc-480c-a7db-071bbeedc9ba",
    "queueName": "",
    "queueId": "",
    "usingRdp": false,
    "message": "",
    "canManage": true,
    "deviceName": "AAIN243FQGYE",
    "userName": "joe-doe-runner",
    "tenantUuid": "b6e4eb84-f7ef-4dfd-a432-725b71de8142",
    "automationPriority": "PRIORITY_MEDIUM",
    "callbackInfo": "",
    "runElevated": false,
    "botLabel": "",
    "currentBotName": ""
  },
  {
    "id": "5e55db1e-c6ec-422b-8718-49224b2f3ce3_03a0f7f8bb373597",
    "automationName": "ButtonBot.2022.05.02.08.07.24.ritesh",
    "fileName": "ButtonBot",
    "filePath": "",
    "type": "RUN_NOW",
    "startDateTime": "2022-05-02T15:07:29.004749Z",
    "endDateTime": "1970-01-01T00:00:00Z",

```



```

    "command": "",
    "status": "UPDATE",
    "progress": 0,
    "automationId": "",
    "userId": "291",
    "deviceId": "161",
    "currentLine": 0,
    "totalLines": 0,
    "fileId": "142853",
    "modifiedBy": "291",
    "createdBy": "291",
    "modifiedOn": "2022-05-02T08:07:31.548640Z",
    "createdOn": "",
    "deploymentId": "05a192ec-967e-47f3-8b81-185a161d9424",
    "queueName": "",
    "queueId": "",
    "usingRdp": false,
    "message": "",
    "canManage": true,
    "deviceName": "form-builder",
    "userName": "johndoe",
    "tenantUuid": "b6e4eb84-f7ef-4dfd-a432-725b71de8142",
    "automationPriority": "PRIORITY_MEDIUM",
    "callbackInfo": "",
    "runElevated": false,
    "botLabel": "",
    "currentBotName": ""
  }
]
}

```

Response parameters

Parameter	Type	Description
offset	Integer	The starting list offset, used for pagination

Parameter	Type	Description
total	Integer	Total number of records
totalFilter	Integer	Number of records after applying the filter
List	Array	The array of List activity objects
List activity object		
id	Integer	The unique ID of a specific activity
automationName	String	Name of the automation
fileName	String	Bot file name
filePath	String	<p>The file path of the bot</p> <div> <p>Note: As filePath is deprecated, use Get Immediate Parents API, endpoint:</p> <pre>GET http://{{ControlRoomURL}}/v2/repository/files/{fileid}/parents</pre> <p>to retrieve the file path based on the input file ID.</p> </div>
type	String	<p>File type associated with the bot. Following are the possible values for <i>Type</i>:</p> <ul style="list-style-type: none"> WORKORDER : Depicts the status of the workitems uploaded from the CSV file WLM_TASK : A Workload management bot EXPORT : Bot export API : A bot that is run from an API

Parameter	Type	Description
		<ul style="list-style-type: none"> • AARI : A process bot • RUN_NOW : A run as a users bot • SCHEDULED : A scheduled bot • TRIGGER : A triggered bot • AUDIT_EXPORT : Audit export • QUEUE_EXPORT : Queue export • QUEUE_IMPORT : Queue import • WORKITEM_EXPORT : List of workitems export in CSV
startDate	String	The date and time of when this bot started
endDate	String	The date and time of when this bot ended
command	String	The current command the bot is on
status	String	<p>The status of the bot. Following values are possible:</p> <ul style="list-style-type: none"> • COMPLETED : Bot successfully completed execution. • DEPLOYED : Autologin is successful, and bot is deployed to a device. • DEPLOY_FAILED : Bot failed to deploy to the device because, for example, autologin failed. • QUEUED : Requested user or device is busy running another execution • PENDING_EXECUTION : Device has been selected, but bot has not yet been deployed to that device • RUNNING or UPDATE : Bot is executing on a device. • RUN_FAILED : Bot failed after being deployed to a device. • RUN_PAUSED : User paused the bot. • RUN_TIMED_OUT : Bot failed to complete tasks within a specific time period.

Parameter	Type	Description
		<ul style="list-style-type: none"> UNKNOWN : Connection between the service and the device was lost.
progress	Integer	The progress of the bot in percentage.
totalLines	Integer	Total number of command lines the bot contains (including the disabled lines).
currentLine	Integer	The current line the bot is processing.
timeTaken	Integer	Time taken in milliseconds by the bot to complete the operation.
progress	Integer	The progress of the bot in percentage.
automationId	Integer	The ID of the automation.
userId	Integer	The ID of the user.
deviceId	Integer	The ID of the device.
fileId	Integer	Unique identifier of the bot file.
modifiedBy	Integer	The ID of the user who modified the activity.
createdBy	Integer	The ID of the user who created the activity.
modifiedOn	String	The timestamp when it was modified.
createdOn	String	The creation timestamp of the activity.
deploymentId	String	The deployment ID of the bot.

Parameter	Type	Description
queueName	String	Name of the queue.
queueId	String	ID of the queue.
usingRdp	Boolean	Flag that shows whether or not the bot is using remote deployment protocol.
message	String	Error message that returns details about the state of the execution.
canManage	Boolean	Flag to show whether the bot can be managed.
deviceName	String	Name of the device.
userName	String	The user name of the user who is running the bot.
tenantUuid	String	The tenant's unique UUID.
automationPriority	String	The automation priority. By default it is set to <code>PRIORITY_MEDIUM</code> . Possible values for <code>automationPriority</code> include: <code>PRIORITY_MEDIUM</code> , <code>PRIORITY_HIGH</code> , and <code>PRIORITY_LOW</code> .
callbackInfo	Object	No value is returned for this parameter; will need individual bot API to retrieve that information
runElevated	Boolean	Flag showing whether the bot was deployed using elevated permissions or not. Possible values include <code>false</code> and <code>true</code> .
botLabel	String	Label for the bot. It can be used to distinguish bots from categories such as production and testing.

Parameter	Type	Description
currentBotName	String	Current name of the bot. It can change during execution.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Parent topic: [Bot Execution Orchestrator API](#)

Bot Scheduler APIs

Use the Bot Scheduler APIs to create, update, delete, and return details on scheduled automations.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

Prerequisites

Ensure that you are allocated the following resources:

- **Schedule my bots to run** feature permission
- **Run and schedule** permissions for the folders that contain the bots.
- Access to the Bot Runner licensed users.
- Access to either a default device or a device pool

Note: If the user associated with the Bot Runner license has a default device assigned to their account, the bot deploys on that device. If no default device is assigned, or you want to select a different device, then you must specify a device pool.

Create an automation schedule

1. [Authenticate the user](#)
2. [List files and folders by workspace API](#)
3. [List available unattended Bot Runners API](#)
4. **Optional:** [List device pools API](#)
5. [Schedule bot to run API](#)

- **[Schedule bot to run API](#)**
Schedule a bot to run on an unattended Bot Runner either one time or on a recurring basis.
- **[List automation schedules API](#)**
Retrieve details of the automation schedules that you have permissions to view.

Parent topic: [Control Room APIs](#)

Schedule bot to run API

Schedule a bot to run on an unattended Bot Runner either one time or on a recurring basis.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- Ensure that you are allocated the following resources:
 - **View and Run my bots** feature permission
 - **Run and schedule** permissions for the folders that contain the bots
 - Access to Bot Runner licensed users
 - Access to either a default device or a device pool

Note: If the user associated with the Bot Runner license has a default device assigned to their account, the bot deploys on that device. If no default device is assigned, or you want to select a different device, then you must specify a device pool.

To schedule a bot, you provide the following information to the API:

Parameter	Required	Type	Description
fileId	Yes	Number	Identifier for the bot. List files and folders by workspace API
runAsUserIds	Yes	Number	Identifier for a user that is registered with your Control Room as an Unattended bot runner. List available unattended Bot Runners API

Parameter	Required	Type	Description
status	Yes	String	Indicates whether to create an active or draft schedule. Enter either <code>ACTIVE</code> or <code>DRAFT</code> .
poolIds	No	Number	Identifier of a device pool that has at least one active device. List device pools API
overrideDefaultDevice	No	Boolean	<p>If the Bot Runner user is assigned to a default device and you want to specify a device pool, set this parameter to <code>true</code>.</p> <p>If deploying to the default device, set this parameter to <code>false</code>.</p>

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST method and endpoint URL: `<control_room_url>/v1/schedule/automations`.

Request body:

This example request body contains the required parameters to schedule a bot to run one time.

```
{
  "name": "UpdateInventory.20.12.16.10.40.48",
  "fileId": 14277,
  "poolIds": [
    "27"
  ],
  "timeZone": "Asia/Calcutta",
  "runAsUserIds": [
    "472",
    "480",
    "80"
  ]
}
```



```
],  
  "startDate":"2020-12-16",  
  "startTime":"15:00",  
  "scheduleType":"NONE",  
  "status":"ACTIVE"  
}
```

This example request body contains the required parameters to schedule a bot to run on a recurring basis.

```
{  
  "name":"UpdateInventory.20.12.16.10.40.48",  
  "fileId":14277,  
  "poolIds":[  
    "27"  
  ],  
  "timeZone":"Australia/Melbourne",  
  "runAsUserIds":[  
    "1103",  
    "36",  
    "80"  
  ],  
  "startDate":"2020-12-16",  
  "repeatOccurrence":{  
    "endTime":"23:59",  
    "runEvery":"1",  
    "timeUnit":"HOURS"  
  },  
  "repeatEnabled":true,  
  "endDate":"2020-12-24",  
  "startTime":"20:45",  
  "weeklyRecurrence":{  
    "interval":"1",  
    "daysOfWeek":[  
      "TUE",  
      "THU",  
      "FRI"  
    ]  
  }  
}
```

```

    ]
  },
  "scheduleType": "WEEKLY",
  "status": "ACTIVE"
}

```

3. Send the request.

Response body: The two example responses include the following information about the automation:

- **id:** the numerical value that identifies the automation. Use this parameter in the Update automations or Delete automations APIs.
- **zonedNextRunDateTime:** the date and time of the next time the bot is scheduled to run.

This is an example response for a bot scheduled to run one time.

```

{
  "id": "989",
  "name": "UpdateInventory.20.12.16.10.40.48",
  "fileId": 14277,
  "status": "ACTIVE",
  "deviceIds": [],
  "description": "",
  "rdpEnabled": false,
  "scheduleType": "NONE",
  "timeZone": "Asia/Calcutta",
  "startDate": "2021-12-16",
  "endDate": "",
  "startTime": "15:00",
  "repeatEnabled": false,
  "zonedNextRunDateTime": "2021-12-16T09:30:00Z",
  "createdBy": "1103",
  "createdOn": "2021-01-11T18:57:18.932407Z",
  "updatedBy": "1103",
  "updatedOn": "2021-01-11T18:57:18.932422Z",
  "tenantId": "1",
  "fileName": "appsheets",
  "filePath": "Automation Anywhere\\Bots\\appsheets",
  "runAsUserIds": [
    "480",

```

```
    "80",
    "472"
  ],
  "botInput": {},
  "tenantUuid": "e100fbce-008c-04ec-4063-7af0af91fb2f",
  "poolId": [
    "11"
  ],
  "overrideDefaultDevice": false,
  "runElevated": false
}
```

This is an example response for a bot scheduled to run on a recurring basis.

```
{
  "id": "990",
  "name": "UpdateInventory.20.12.16.10.40.48",
  "fileId": 14277,
  "status": "ACTIVE",
  "deviceIds": [],
  "description": "",
  "rdpEnabled": false,
  "scheduleType": "WEEKLY",
  "weeklyRecurrence": {
    "interval": 1,
    "daysOfWeek": [
      "TUE",
      "THU",
      "FRI"
    ]
  },
  "timeZone": "Australia/Melbourne",
  "startDate": "2021-01-16",
  "endDate": "2021-12-24",
  "startTime": "20:45",
  "repeatEnabled": true,
  "repeatOccurrence": {
```

```

    "runEvery": "1",
    "timeUnit": "HOURS",
    "endTime": "23:59"
  },
  "zonedNextRunDateTime": "2021-01-19T09:45:00Z",
  "createdBy": "1103",
  "createdOn": "2021-01-11T18:59:31.182663Z",
  "updatedBy": "1103",
  "updatedOn": "2021-01-11T18:59:31.182669Z",
  "tenantId": "1",
  "fileName": "a_trigger",
  "filePath": "Automation Anywhere\\Bots\\a_trigger",
  "runAsUserIds": [
    "80",
    "36",
    "1103"
  ],
  "botInput": {},
  "tenantUuid": "e100fbce-008c-04ec-4063-7af0af91fb2f",
  "poolId": [
    "7"
  ],
  "overrideDefaultDevice": false,
  "runElevated": false
}

```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Next steps

Use the [List automation schedules API](#) to retrieve details on all the scheduled automations.

Parent topic: [Bot Scheduler APIs](#)

List automation schedules API

Retrieve details of the automation schedules that you have permissions to view.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- You must be assigned a role that includes the `View and manage all scheduled activity from my Folders` permission.

In this example, you list all the scheduled automations sorted by the next run date and time.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST method and endpoint URL: `<control_room_url>/v1/schedule/automations/list`.

Request body:

```
{
  "sort": [
    {
      "field": "zonedNextRunDateTime",
      "direction": "asc"
    }
  ]
}
```

3. Send the request.

Response body: The response includes the following data about each automation:

- **id:** the numerical value that identifies the automation. Use this parameter in the Update automations or Delete automations APIs.
- **status:** returns whether the scheduled automation is `ACTIVE` or `INACTIVE`.

```
{
  "page": {
    "offset": 0,
    "total": 3,
  }
}
```

```
    "totalFilter": 3
  },
  "list": [
    {
      "id": "661",
      "name": "eodReport_28.20.10.12.23.02.48",
      "fileId": 6598,
      "status": "ACTIVE",
      "deviceIds": [],
      "description": "",
      "rdpEnabled": false,
      "scheduleType": "DAILY",
      "dailyRecurrence": {
        "interval": 3
      },
      "timeZone": "America/Los_Angeles",
      "startDate": "2020-10-12",
      "endDate": "",
      "startTime": "23:30",
      "repeatEnabled": true,
      "repeatOccurrence": {
        "runEvery": "3",
        "timeUnit": "HOURS",
        "endTime": "23:59"
      },
      "zonedNextRunDateTime": "2021-01-14T07:30:00Z",
      "createdBy": "251",
      "createdOn": "2020-10-13T06:01:51.992433Z",
      "updatedBy": "251",
      "updatedOn": "2021-01-11T07:30:00.082057Z",
      "tenantId": "1",
      "fileName": "wlm_28",
      "filePath": "Automation Anywhere\\Bots\\West Coast",
      "runAsUserIds": [
        "251"
      ],
    },
  ],
```

```
"botInput": {},
"tenantUuid": "e100fbce-008c-04ec-4063-7af0af91fb2f",
"poolId": [
  "23"
],
"overrideDefaultDevice": false,
"runElevated": false
},
{
  "id": "990",
  "name": "accounts.20.12.16.10.51.59",
  "fileId": 12501,
  "status": "ACTIVE",
  "deviceIds": [],
  "description": "",
  "rdpEnabled": false,
  "scheduleType": "WEEKLY",
  "weeklyRecurrence": {
    "interval": 1,
    "daysOfWeek": [
      "TUE",
      "THU",
      "FRI"
    ]
  },
  "timeZone": "Australia/Melbourne",
  "startDate": "2021-01-16",
  "endDate": "2021-12-24",
  "startTime": "20:45",
  "repeatEnabled": true,
  "repeatOccurrence": {
    "runEvery": "1",
    "timeUnit": "HOURS",
    "endTime": "23:59"
  },
  "zonedNextRunDateTime": "2021-01-19T09:45:00Z",
```

```
"createdBy": "1103",
"createdOn": "2021-01-11T18:59:31.182663Z",
"updatedBy": "1103",
"updatedOn": "2021-01-11T18:59:31.182669Z",
"tenantId": "1",
"fileName": "a_trigger",
"filePath": "Automation Anywhere\\Bots\\APAC",
"runAsUserIds": [
  "80",
  "36",
  "1103"
],
"botInput": {},
"tenantUuid": "e100fbce-008c-04ec-4063-7af0af91fb2f",
"poolId": [
  "7"
],
"overrideDefaultDevice": false,
"runElevated": false
},
{
  "id": "989",
  "name": "UpdateInventory.20.12.16.10.40.48",
  "fileId": 11201,
  "status": "ACTIVE",
  "deviceIds": [],
  "description": "",
  "rdpEnabled": false,
  "scheduleType": "NONE",
  "timeZone": "Asia/Calcutta",
  "startDate": "2021-12-16",
  "endDate": "",
  "startTime": "15:00",
  "repeatEnabled": false,
  "zonedNextRunDateTime": "2021-12-16T09:30:00Z",
  "createdBy": "1103",
```



```

    "createdOn": "2021-01-11T18:57:18.932407Z",
    "updatedBy": "1103",
    "updatedOn": "2021-01-11T18:57:18.932422Z",
    "tenantId": "1",
    "fileName": "appsheets",
    "filePath": "Automation Anywhere\\Bots\\spreadsheets",
    "runAsUserIds": [
      "480",
      "80",
      "472"
    ],
    "botInput": {},
    "tenantUuid": "e100fbce-008c-04ec-4063-7af0af91fb2f",
    "poolId": [
      "11"
    ],
    "overrideDefaultDevice": false,
    "runElevated": false
  }
]
}

```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Parent topic: [Bot Scheduler APIs](#)

Repository Management APIs

Use the Repository Management APIs to return information on or to delete the objects (bots, folders, and files) that you have permissions to access in the Control Room.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The Automation 360 repository is comprised of the private and public workspaces, which contain folders of bots and their dependency files. The private workspace holds objects that are only accessible to the logged-in user. Bots in the private workspace can only be run on that user's device. The public workspace holds objects that are accessible to all users who have the necessary permissions. When a bot is checked-in from a

private workspace to the public workspace, it can be checked out by another user to their private workspace for editing or deployed to a Bot Runner.

Note:

- To view objects in your private workspace, you must have a Bot Creator license and a role that includes the **View my bots** feature permission.
- To view objects in the public workspace, you must be assigned a role that includes the **View my bots** feature permission and the **View content** bot permission to folders in the public workspace.

Choosing a Repository Management list API

The following Repository Management APIs return information on objects (bots, files, and folders), based on the access granted by the user's role.

- `/file/list` returns details on all the objects for which you have access permissions.

[List files API](#)

- `/folders/{folderid}/list` returns details on only the objects in a specific folder.

[List files and folders in a specific folder API](#)

- `/workspaces/{workspaceType}/files/list` returns details on only the objects in either the public or private workspace.

[List files and folders by workspace API](#)

Supported filterable fields

Use the following filters in the request bodies of the list APIs to narrow down the response data.

createdBy

The numeric identifier for the user who created a folder or bot.

```
{
  "operator": "eq",
  "field": "createdBy",
  "value": "2587"
}
```

folder

This example searches for only folders. Set the value to `false` to search for only bots and files.

```
{
  "operator": "eq",
  "field": "folder",
  "value": "true"
}
```

name

This example searches for objects that are named **Finance** or **finance**. This search is not case-sensitive.

- **Field:** name
- **Type:** string

```
{
  "filter": {
    "operator": "eq",
    "value": "finance",
    "field": "name"
  }
}
```

path

This example searches for objects that contain the string **Finance** in the path parameter. This search is not case-sensitive.

- **Field:** path
- **Type:** string

```
{
  "filter": {
    "operator": "substring",
    "value": "Finance",
    "field": "path"
  }
}
```

- **List files API**

Use the List files API to view the details of all the objects (file, folder, or bot) in the Control Room. This API returns an id parameter as the response, which is a numeric value that can be used in different APIs to identify the file, folder, or bot.

- **List files and folders in a specific folder API**

Return details about objects (bots, folders, and files) in a specific parent folder. This endpoint returns the object id, which is a numeric value that is used in other APIs to identify the file, folder or bot.

- **List files and folders by workspace API**

Return details on objects (files, folders, and bots) in either the public or private workspace. This endpoint returns the object id, which is a numeric value that is used in other APIs to identify the file, folder or bot.

- **Get Immediate Parents API**

Use this API to get the immediate parents of a Task Bot. The Get Immediate Parents API gets a *fileId* as input and lists the immediate parent details.

- **Delete file/folder API**

Use this API to delete objects (bots, files, or folders) from the public or your private workspace.

Parent topic: [Control Room APIs](#)

List files API

Use the List files API to view the details of all the objects (file, folder, or bot) in the Control Room. This API returns an id parameter as the response, which is a numeric value that can be used in different APIs to identify the file, folder, or bot.

Note: The List files API will be deprecated from version Automation 360 v.29. This API has the following limitations:

- If you use it as an AAE_Admin, you can view bots from the private repository in the response.
- If you use it as a bot creator, you can view system bots (recorder) in the response.

As a workaround, you can use the [List files and folders by workspace API](#).

Request

```
POST http://{{ControlRoomURL}}/v2/repository/file/list
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

- To view objects in your private workspace, you must have a Bot Creator license and a role that includes the **View my bots** feature permission.
- To view objects in the public workspace, you must be assigned a role that includes the **View my bots** feature permission and the **View content** bot permission to folders in the public workspace.

You can send an API request with or without filter parameters. An API request without any filter parameters specified retrieves the details of all the objects in the Control Room. You can use the filter parameters to retrieve a specific set of file or folder objects instead of fetching all objects in the Control Room. For more information, see [Filtering, pagination, and sorting](#).

Request body without filters:

```
{
  "filter": null,
  "sort": [
    {
      "field": "directory",
      "direction": "desc"
    },
    {
      "field": "name",
      "direction": "asc"
    }
  ],
  "page": {
    "offset": 0,
    "length": 100
  }
}
```

Request body with filters:

```
{
  "filter": {
    "operator": "substring",
    "value": "bot",
  }
}
```

```

    "field": "name"
  },
  "sort": [
    {
      "field": "directory",
      "direction": "desc"
    },
    {
      "field": "name",
      "direction": "asc"
    }
  ],
  "page": {
    "offset": 0,
    "length": 100
  }
}

```

Request parameters

Parameter	Type	Required	Description
filter	Object	No	<p>Filters the result based on operator, field, or value.</p> <p>operator</p> <p>Allowed enumerations are NONE, lt, le, eq, ne, ge, gt, substring, and, or, not.</p> <p>field</p> <p>Allowed values are name, lastModified, path, or folder.</p> <p>value</p> <p>Specify a value for the name, lastModified, path, or folder that you have selected in the field parameter.</p>
sort	Array	No	<p>By default, search results are sorted in descending order of their IDs. To specify an alternative sorting, use the sort query parameter.</p>

Parameter	Type	Required	Description
			Enter the field by which you want to sort along with the direction asc (ascending) or desc (descending).
page	Object	No	The page object allows you to get the desired pages.

Response

```
{
  "page": {
    "offset": 0,
    "total": 4619,
    "totalFilter": 100
  },
  "list": [
    {
      "id": "137169",
      "parentId": "111492",
      "name": ".25Bot",
      "permission": {
        "delete": true,
        "download": true,
        "upload": true,
        "run": true,
        "publishBotstore": false
      },
      "lastModified": "2022-04-11T10:28:18.098323Z",
      "lastModifiedBy": "291",
      "path": "Automation Anywhere\\Bots\\.25Bot",
      "directory": false,
      "size": "2615",
      "locked": false
    }
  ]
}
```

```
}
  ] }
```

Response parameters

Parameter	Type	Description
offset	Integer	The starting list offset, used for pagination.
total	Integer	Total number of records.
totalFilter	Integer	Number of records after applying the filter.
List	Array	The list of directories and files.
List objects		
id	Integer	The unique ID of the displayed objects (bots, folders, or files).
parentId	Integer	The unique ID of the parent folder.
name	String	Name of the file or folder.
permission	Object	<p>Displays permissions for the current user with the values True or False:</p> <p>delete Indicates whether the current logged-in user has rights to delete the bot.</p> <p>download Indicates whether the current logged-in user has rights to download (check out) the bot.</p> <p>upload Indicates whether the current logged-in user has rights to upload (check in) the bot.</p> <p>run Indicates whether the current logged-in user has rights to run or schedule the bot.</p>

Parameter	Type	Description
		publishBotstore Indicates whether the current logged-in user has rights to publish the bot to the bot store.
lastModified	Integer	Date and time when the bot was last updated.
lastModifiedBy	String	ID of the user who last updated the bot or file.
path	Integer	Path of the file or folder in the repository.
directory	String	Flag for directory.
size	Integer	Size of the file. It is available only if the item type is file.
locked	String	Indicates whether the file is locked. It is available only if the item type is file.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Parent topic: [Repository Management APIs](#)

List files and folders in a specific folder API

Return details about objects (bots, folders, and files) in a specific parent folder. This endpoint returns the object id, which is a numeric value that is used in other APIs to identify the file, folder or bot.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- To view objects in your private workspace, you must have a Bot Creator license and a role that includes the **View my bots** feature permission.
- To view objects in the public workspace, you must be assigned a role that includes the **View my bots** feature permission and the **View content** bot permission to folders in the public workspace.
- You require the folder ID for the folder you want to search in. Use one of the following Repository Management APIs to retrieve the object ID:
 - [List files API](#)
 - [List files and folders by workspace API](#)

The example in this task searches for subfolders that contain the string **finance**.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST method and endpoint URL: `<control_room_url>/v2/repository/folders/{folderId}/list`

{folderId} is the object ID of the folder that in which you want to search.

Request body: The following example request searches for folders that contain the word **finance** in the name.

```
{
  "filter": {
    "operator": "and",
    "operands": [
      {
        "operator": "substring",
        "field": "name",
        "value": "finance"
      },
      {
        "operator": "eq",
        "field": "folder",
        "value": "true"
      }
    ]
  }
}
```

```
}
}
```

[Supported filterable fields](#)

3. Send the request.

Response body: In a successful request, this endpoint returns the following data:

- **id**: a unique numeric identifier for the object that matches the search parameters.
- **parentId**: a unique numeric identifier for the parent folder.
- **folder**: a boolean value that returns **true** if the object is a folder and **false** if it is a bot or other file.

In this example response, the endpoint returns a folder with the object **id** of 40378.

```
{
  "page": {
    "offset": 0,
    "total": 329,
    "totalFilter": 1
  },
  "list": [
    {
      "id": "40378",
      "parentId": "2",
      "name": "EOD-finance-reports",
      "path": "Automation Anywhere\\Bots\\APAC\\EOD-finance-reports",
      "description": "",
      "type": "application/vnd.aa.directory",
      "size": "0",
      "folder": true,
      "folderCount": "0",
      "productionVersion": "",
      "latestVersion": "",
      "locked": false,
      "lockedBy": "0",
      "createdBy": "2587",
      "lastModifiedBy": "2587",
      "lastModified": "2020-09-02T05:26:51.162916Z",
      "permission": {
```

```

        "delete": true,
        "download": false,
        "upload": false,
        "run": true,
        "publishBotstore": false,
        "viewContent": true,
        "clone": false,
        "editContent": true,
        "createFolder": true,
        "move": true,
        "cancelCheckout": false,
        "revertCheckout": false
    },
    "workspaceId": "0",
    "botStatus": "DRAFT",
    "hasErrors": false,
    "workspaceType": "UNKNOWN",
    "metadata": false,
    "uri": "",
    "version": "0",
    "hasTriggers": false
}
]
}

```

Use the numeric identifier, such as an id in subsequent APIs.

Parent topic: [Repository Management APIs](#)

List files and folders by workspace API

Return details on objects (files, folders, and bots) in either the public or private workspace. This endpoint returns the object id, which is a numeric value that is used in other APIs to identify the file, folder or bot.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- To view objects in your private workspace, you must have a Bot Creator license and a role that includes the `View my bots` feature permission.
- To view objects in the public workspace, you must be assigned a role that includes the `View my bots` feature permission and the `View content` bot permission to folders in the public workspace.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST method and endpoint URL: `<control_room_url>/v2/repository/workspaces/{workspaceType}/files/list`
{workspaceType} specifies whether to search in the public or private workspace.

The following example request searches for objects that contain the string **finance** in the name.

Request body:

```
{
  "filter": {
    "operator": "substring",
    "field": "name",
    "value": "finance"
  }
}
```

[Supported filterable fields](#)

3. Send the request.

Response body:

In this example response, this endpoint returns a bot with the object `id` of 14277.

```
{
  "page": {
    "offset": 0,
    "total": 1114,
    "totalFilter": 1
  },
  "list": [
```

```
{
  "id": "14277",
  "parentId": "9",
  "name": "financeWeeklyReport",
  "path": "Automation Anywhere\\Bots\\exampleBots",
  "description": "v1",
  "type": "application/vnd.aa.taskbot",
  "size": "799",
  "folder": false,
  "folderCount": "0",
  "productionVersion": "",
  "latestVersion": "",
  "locked": false,
  "lockedBy": "0",
  "createdBy": "22",
  "lastModifiedBy": "22",
  "lastModified": "2020-10-21T17:42:10.140037Z",
  "permission": {
    "delete": false,
    "download": false,
    "upload": false,
    "run": true,
    "publishBotstore": false,
    "viewContent": false,
    "clone": false
  },
  "workspaceId": "0",
  "botStatus": "PUBLIC",
  "hasErrors": false,
  "workspaceType": "UNKNOWN",
  "metadata": false,
  "uri": "",
  "version": "3",
  "hasTriggers": false,
  "isModified": false
}
```

```
]
}
```

Response parameters:

- **id** : a unique numeric identifier for the object that matches the search parameters.
- **parentId** : a unique numeric identifier for the parent folder.
- **folder** : a boolean value that returns **true** if the object is a folder and **false** if it is a bot or other file.

The REST API responds to each request with an HTTP response code. For details about the response codes, see [API response codes](#).

Next steps

If you are performing the steps to run a bot or to create an automation schedule, perform this task: [List available unattended Bot Runners API](#)

Parent topic: [Repository Management APIs](#)

Get Immediate Parents API

Use this API to get the immediate parents of a Task Bot. The Get Immediate Parents API gets a *fileId* as input and lists the immediate parent details.

Request

```
GET http://{{ControlRoomURL}}/v2/repository/files/{fileid}/parents
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request Parameters

Parameter	Type	Required	Description
fileid	Integer	Yes	File id for which you want to view the parents.

```
GET http://{{ControlRoomURL}}/v2/repository/files/698/parents
```

Response

```
{
  "dependencies":[
    {
      "id":"349",
      "name":"Parent",
      "path":"Automation Anywhere\\Bots\\demo_test\\Parent",
      "size":"1536",
      "type":"application/vnd.aa.taskbot",
      "dependencyType":"SCANNED",
      "url":"","
      "requiredByFileId":"349",
      "version":"0",
      "lockedBy":"0",
      "botStatus":"PUBLIC",
      "permission":{
        "delete":true,
        "download":true,
        "upload":true,
        "run":true,
        "publishBotstore":false,
        "viewContent":true,
        "clone":true,
        "editContent":false,
        "createFolder":false,
        "move":false,
        "cancelCheckout":false,
        "revertCheckout":false,
        "viewHistory":false,
        "labelBots":false
      },
      "versionNumber":"0",
      "label":""
    }
  ]
}
```



```

    }
  ]
}
```

Response Parameters

Parameter	Type	Description
id	Integer	File Id of the parent.
name	String	Name of the parent.
path	String	Path of the parent file.
size	Integer	Size of the parent file.
type	String	Type of the parent.
dependencyType	String	<p>The type of the related object.</p> <ul style="list-style-type: none"> • NONE: If the bot or file does not have a parent, then the API will return the same bot or file. • SCANNED: It is the relation with the child bots or files which are scanned and found automatically. • MANUAL: It is the relation with the bots or files that are added to the parent bot manually.
url	String	Not pertinent to this API.
requiredByFileId	Integer	The Id of the parent bot that is calling the child bot.
version	Integer	Version of the parent file.
lockedBy	Integer	The user that has checked out this file.

Parameter	Type	Description
botStatus	String	bot status. Possible values can be: NEW, CHECKED_OUT, CLONED, and PUBLIC.
permission	Object	Lists all the permissions of the parent.

Parent topic: [Repository Management APIs](#)

Delete file/folder API

Use this API to delete objects (bots, files, or folders) from the public or your private workspace.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- You can delete bots, files, or folders from your private workspace. To delete objects from the public workspace, you must be assigned a role that contains the **Delete** bot permission on the folder that contains the objects that you want to delete.
- If you are deleting a folder, ensure that it is empty. Only empty folders can be deleted.
- To delete a file or folder, you must provide the object ID. Use one of the Repository Management list APIs to retrieve the object ID. [Choosing a Repository Management list API](#)

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the DELETE method and endpoint URL: `<control_room_URL>/v2/repository/files/{id}`
{id} is the object id of the file or folder that you want to delete.
3. Send the request.

Response body: A successful request returns the following message

```
204 Successful delete
```

The REST API responds to each request with an HTTP response code. For details about the response codes, see [API response codes](#).

Parent topic: [Repository Management APIs](#)

Bot Insight API

Users with the `AAE_Bot Insight Admin` or `AAE_Admin` role and the `Bot Insight` license can access the Bot Insight API to retrieve business and operations data.

Automation Anywhere bots are built, run, and monitored in the Control Room. Bot Insight accesses real-time business insights and digital workforce performance data to use content-level productivity data from the bots that are deployed.

Business data

The Business data endpoints return Bot Insight data retrieved from deployed bots.

- [Delete task log data](#)

Delete the business data that is logged in the Bot Insight database on a deployed bot.

```
DELETE /v2/botinsight/data/api/deletetasklogdata
```

- [Get bot variables data](#)

Retrieve information about the variables in deployed bots, such as the variable name, data type, minimum value, and maximum value.

```
GET /v2/botinsight/data/api/gettaskvariableprofile
```

- [Get task log data](#)

Use the get task log data to retrieve the analytical variables data that is logged during a bot run.

```
GET /v2/botinsight/data/api/gettasklogdata
```

Operations data

The Operations data endpoints return information about bots that were deployed to Bot Runner devices. You can use this information to enhance productivity and take measures based on real-time information for RPA deployments.

- [Get audit trail data](#)

Retrieve information about Control Room events.

```
GET /v2/botinsight/data/api/getaudittraildata
```

- [Get bot run data](#)

Retrieve information about a bot run, such as the server information and whether it ran successfully or encountered an error.

```
GET /v2/botinsight/data/api/getbotrundata
```

Get task log data

Use the get task log data to retrieve the analytical variables data that is logged during a bot run.

Request

```
GET https://{{ControlRoomURL}}/v2/botinsight/data/api/gettasklogdata?botname=ATMReconciliation&fromdate=2022-07-04T00%3A00%3A00&todate=2022-07-07T23%3A59%3A59&limit=100&pageno=1
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

The URL includes the example query parameter botname. For large datasets, use the pageno and limit parameters to avoid a timeout error.

- You must have the **AAE_Bot Insight Admin** role and the **Bot Insight** license.
- The bot must run at least for one time with the Bot Creator (Private workspace).
- The bot must be checked in and deployed to the public workspace.

Request Parameters

Parameter	Type	Required	Description
botname	string	No	<p>Name of the bots for which you retrieve data</p> <p>Enter up to 10 bot names and separate each name with a comma.</p> <p>If you do not provide this parameter, the API will return data on all the bots.</p>
pageno	integer	No	Page number from which to retrieve the data
limit	integer	No	<p>Specifies the number of parts in which the information is retrieved</p> <p>For example, if you specify the limit as 2500 to retrieve a total of 10000 records, then the information retrieved is as follows:</p> <ul style="list-style-type: none"> • 0 page returns 1 - 2500 • 1st page returns 2501 - 5000 • 2nd page returns 5001 - 7500 • 3rd page returns 7501 - 10000 • Min value: 1
fromdate	date	No	<p>Start date of the period for which to retrieve the data</p> <p>If you do not provide this parameter, the API will return all available data.</p> <p>Format: yyyy-mm-ddThh:mm:ss .</p>
todate	date	No	<p>End date of the period for which to retrieve the data</p> <ul style="list-style-type: none"> • Format: yyyy-mm-ddThh:mm:ss

Parameter	Type	Required	Description
			<ul style="list-style-type: none"> Default: current date

Note: View the migration status using the [List migration results API](#).

Response

This response example contains data on the *ATMReconciliation* bot and the first record returned.

```
{
  "businessData": [
    {
      "totalRecords": 1,
      "count": 1,
      "pageNo": 1,
      "botId": "PROD_581",
      "botName": "ATMReconciliation",
      "repositoryPath": "repository:///Automation%20Anywhere/Bots/Gettas
klogdata?fileId=580&workspace=PRIVATE&version=0&label=",
      "list": [
        {
          "transactionName": "Default",
          "transactions": [
            {
              "runId": "e29732ad-5339-4012-941b-e9a1eb47806c_1a8
2fe211fc8865e",
              "transactionId": "2bda3a08-3049-4147-9e2b-8dc9dd8d
7665",
              "dateLogged": "2022-07-06T10:33:11",
              "variables": {
                "variable1": "123.0",
                "variable3": "789.0",
                "variable2": "456.0"
              }
            }
          ]
        }
      ]
    }
  ]
}
```

```

    }
  ]
}
]
}
]
}
]
}

```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Response Parameters

Parameter	Type	Description
runId	String	Identifier for the bot run that retrieved this data. All the transactions in a single run have a common <code>runId</code> .
transactionId	String	Identifier for the data set that was retrieved in a single loop iteration.
dateLogged	date	Date and time the bot retrieved the data. Format: <code>yyyy-mm-ddThh:mm:ss</code>
variables	Any	Variable names and values.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

Delete task log data

Delete the business data that is logged in the Bot Insight database on a deployed bot.


Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- You must have the **AAE_Bot Insight Admin** role and the **Bot Insight** license.

To delete data from the Bot Insight database, you provide the following information to the API in the request body:

Parameter	Required	Type	Description
<code>botname</code>	yes	string	Name of the bot for which you retrieve data
<code>repositorypath</code>	yes	string	Control Room repository path of the bot
<code>environment</code>	yes	string	Specifies the environment: <code>DEV</code> or <code>PROD</code>
<code>runId</code>	no	string	<p>ID number that is generated when the bot runs.</p> <p>Use the get task log data to retrieve the runId: Get task log data.</p>
<code>fromDate</code>	no	date	<p>Start date of the period for which to retrieve the data</p> <p>Format: <code>yyyy-mm-ddThh:mm:ss</code></p> <p>If you do not provide this parameter, the API will return all available data.</p> <div> <p>Note: Do not provide this parameter if you provide the <code>runId</code> parameter.</p> </div>
<code>toDate</code>	no	date	<p>End date of the period for which to retrieve the data</p> <ul style="list-style-type: none"> • Format: <code>yyyy-mm-ddThh:mm:ss</code>

Parameter	Required	Type	Description
			<ul style="list-style-type: none"> Default: current date <div>  Note: Do not provide this parameter if you provide the <code>runId</code> parameter. </div>

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the DELETE method and endpoint URL: `<control_room_url>/v2/botinsight/data/api/deletetasklogdata`

Request body:

```
{
  "botName": "AnalyticsMortgageProcessing",
  "runId": "a4e706f2-6806-49eb-8d8f-4b915f9a67b0_aaa8b68b1ef888a0",
  "repositoryPath": "repository:///Automation%20Anywhere/Bots/folder8092/AnalyticsMortgageProcessing?fileId=40642&workspace=PRIVATE",
  "environment": "Prod"
}
```

3. Send the request.

Response body: This example response returns the one thousand rows of data that were deleted.

```
{
  "botName": "AnalyticsMortgageProcessing",
  "repositoryPath": "repository:///Automation%20Anywhere/Bots/folder8092/AnalyticsMortgageProcessing?fileId=40642&workspace=PRIVATE",
  "deleteCount": 1000
}
```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Get bot run data

Retrieve information about a bot run, such as the server information and whether it ran successfully or encountered an error.

Request

```
GET http://{{ControlRoomURL}}/v2/botinsight/data/api/getbotrundata?pageno={pageno}&limit={limit}&fromDate={fromDate}&toDate={toDate}
```

Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- To view the bot run data, you must have one of the following:
 - **AAE_BOT_INSIGHT_ADMIN** role
 - **ANALYTICSCLIENT_VALUE** license along with **AAE_ADMIN** role

Request parameters

Parameter	Type	Required	Description
pageno	Integer	No	The page number of the Bot Run Data that you want to retrieve.
limit	Integer	No	<p>Limits the number of results returned. Defaults to 1000.</p> <p>For example, if you specify the limit as 2500 to retrieve a total of 10000 records, then the information retrieved is as follows:</p> <ul style="list-style-type: none"> • 0 page returns 1 - 2500 • 1st page returns 2501 - 5000 • 2nd page returns 5001 - 7500 • 3rd page returns 7501 - 10000
fromDate	String	No	The starting timestamp for the date range.

Parameter	Type	Required	Description
			<p>If you do not provide this parameter, the API will return all available data.</p> <p>Format: <code>yyyy-mm-ddThh:mm:ss</code></p>
toDate	String	No	<p>The ending timestamp for the date range.</p> <ul style="list-style-type: none"> Format: <code>yyyy-mm-ddThh:mm:ss</code> Default: current date

```
GET http://{{ControlRoomURL}}/v2/botinsight/data/api/getbotrundata?pageno=1&limit=2&fromDate=2022-01-27T00%3A30%3A00Z&toDate=2022-02-27T06%3A30%3A00Z
```

Response

```
{
  "totalRecords":2,
  "botRunDataList":[
    {
      "id":60,
      "userName":"runner",
      "hostName":"WIN-MT6N77BI0C2",
      "fileName":"newtestbot27",
      "fileType":"RUN_NOW",
      "startDate":"2022-01-27T06:36:36Z",
      "endDate":"2022-01-27T06:36:36Z",
      "status":"FAILED",
      "totalLines":3,
      "currentLine":3,
      "timeTaken":1852,
      "progress":100
    },
    {
```

```

      "id":30,
      "userName":"runner",
      "hostName":"WIN-MT6N77BI0C2",
      "fileName":"Blm_test_bot",
      "fileType":"RUN_NOW",
      "startDate":"2022-01-28T11:33:25Z",
      "endDate":"2022-01-28T11:33:26Z",
      "status":"COMPLETED",
      "totalLines":3,
      "currentLine":3,
      "timeTaken":1820,
      "progress":100
    }
  ]
}

```

Response Parameters

Parameter	Type	Description
totalRecords	Integer	Total Records that has been retrieved as part of the request.
botRunDataList	Array	The array of Bot Run Data objects.
botRunDataList Object		
id	Integer	The bot run unique Id for the particular run.
userName	String	The user name of the user who is running the bot.
hostName	String	The host name.
fileName	String	Bot file name.
fileType	String	File Type associated to this bot. Following are the possible values for the <i>fileType</i> :

Parameter	Type	Description
		<ul style="list-style-type: none"> WORKORDER : Depicts the status of the workitems uploaded from the CSV file WLM_TASK : A Workload management bot EXPORT : Bot export API : A bot that is run from an API AARI : A process bot RUN_NOW : A run as a users bot SCHEDULED : A scheduled bot TRIGGER : A triggered bot AUDIT_EXPORT : Audit export QUEUE_EXPORT : Queue export QUEUE_IMPORT : Queue import WORKITEM_EXPORT : List of workitems export in CSV
startDate	String	The date and time of when this bot started.
endDate	String	The date and time of when this bot ended.
status	String	<p>The status of the bot. Following values are possible:</p> <ul style="list-style-type: none"> COMPLETED : bot successfully completed execution. DEPLOYED : auto-login is successful and bot is deployed to a device. DEPLOY_FAILED : bot failed to deploy to the device. For example, if auto-login failed. QUEUED : requested user or device is busy running another execution. PENDING_EXECUTION : device has been selected, but bot has not yet been deployed to that device. RUNNING or UPDATE : bot is executing on a device. RUN_FAILED : bot failed after being deployed to a device. RUN_PAUSED : user paused the bot. RUN_TIMED_OUT : bot failed to complete tasks within a specific time period.

Parameter	Type	Description
		<ul style="list-style-type: none"> UNKNOWN : connection between the service and the device was lost.
totalLines	Integer	Total number of command lines the bot contains.
currentLine	Integer	The current line the bot is processing.
timeTaken	Integer	Time taken in milliseconds by the bot to complete the operation.
progress	Integer	The progress of the bot in percentage.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Get Bot Insight audit trail data

Retrieve information about Control Room events.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- You must have the **AAE_Admin** role or a custom role with the **View everyone's audit log actions** permission.

To retrieve information about Control Room events, you provide the following query parameters to the API:

Parameter	Required	Type	Description
<code>pageno</code>	no	integer	Page number from which to retrieve the data
<code>limit</code>	no	integer	<p>Specifies the number of parts in which the information is retrieved</p> <p>For example, if you specify the limit as <code>2500</code> to retrieve a total of 10000 records, then the information retrieved is as follows:</p> <ul style="list-style-type: none"> • 0 page returns 1 - 2500 • 1st page returns 2501 - 5000 • 2nd page returns 5001 - 7500 • 3rd page returns 7501 - 10000
<code>fromDate</code>	no	date	<p>Start date of the period for which to retrieve the data</p> <p>If you do not provide this parameter, the API will return all available data.</p> <p>Format: <code>yyyy-mm-ddThh:mm:ss</code></p>
<code>toDate</code>	no	date	<p>End date of the period for which to retrieve the data.</p> <ul style="list-style-type: none"> • Format: <code>yyyy-mm-ddThh:mm:ss</code> • Default: current date

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the GET method and endpoint URL: `<control_room_URL>/v2/botinsight/data/api/getaudittraildata`
3. Send the request.

Response body:

```
{
  "totalRecords": 10000,
  "auditTrailDataList": [
    {
      "activityType": "BI_LOAD_DASHBOARD",
      "createdBy": 1121,
      "createdOn": "2021-01-11T11:15:06Z",
      "detail": {},
      "environmentName": "DEV",
      "eventDescription": "BI Load Dashboard",
      "hostName": "192.xxx.xxx.xxx",
      "id": "AnalyticsMortgageProcessing",
      "objectName": "N/A",
      "requestId": "f36e040e-02a6-4fae-8415-9ff57067b7a3",
      "source": "Bot Insight",
      "status": "Successful",
      "userName": "ram"
    },
    // The remaining data is omitted from this code example.
  ]
}
```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Get bot variables data

Retrieve information about the variables in deployed bots, such as the variable name, data type, minimum value, and maximum value.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- You must have the **AAE_Bot Insight Admin** role and the **Bot Insight** license.
- The bot must be checked into the public workspace.
- You must publish at least one dashboard to get the results using this API.

To retrieve information about the variables in a deployed bot, you provide the following query parameters to the API:

Parameter	Required	Type	Description
botname	no	string	Name of the bot for which data is retrieved
repositorypath	no	string	Control Room repository path of the bot

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the GET method and endpoint URL: `<control_room_URL>/v2/botinsight/data/api/gettaskvariableprofile`
3. Send the request.

Response body: This response example contains data on the AnalyticsMortgageProcessing bot and the first record returned. To keep this block of code short, the remaining data is omitted.

```
{
  "dataProfiles": [
    {
      "botId": "PROD_40642",
      "botName": "AnalyticsMortgageProcessing",
      "standardDashboardName": "",
      "profileVariables": [
        {
          "variableName": "state",
          "displayName": "State",
          "attributeType": "US_STATE_CODE",
          "sumOfValue": 0.0,
          "minimumValue": "",
          "maximumValue": "",
          "averageOfValues": 0.0,
          "totalDistinct": "53",
          "enabled": "Y",
          "isTransactionVariable": ""
        }
      ]
    }
  ]
}
```

```
    },  
    // The remaining data is omitted from this code example.
```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Bot Lifecycle Management API

Use the Bot Lifecycle Management API to export and import bots with dependent files and command packages for comprehensive automation lifecycle management. Users can export bots from public workspace and import to a private workspace in another Control Room and check into a public workspace.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

Overview

You can use the Control Room Export and Import REST API to manage your automation TaskBots, including dependent files in different environments such as development, testing, and production based on your organization's automation requirements.

For example, you can move bots that are verified as production-ready from test to production.

Import Enterprise 11 bots in Automation 360 for migration

The Bot Lifecycle Management import feature also support to import your Enterprise 11 bots from Enterprise 11 Control Room instance to Automation 360. This enables you to consolidate the Enterprise 11 bots from multiple Control Room repository in a single Automation 360 repository.

Permissions required to move bots

To export bots from the Control Room, you must have the following permissions:

- **Export bot** and **View packages** feature permissions
- **Check in** or **Check out** folder permissions

To import bots to the Control Room, you must have the following permissions:

- **Import bot** and **Manage package** feature permissions
- **Check in** folder permission
- Bot Creator license

Moving a bot

To move a bot from one environment to another, follow these steps:

1. Use the Export API to export the bot from the Control Room in the source environment.

[Export files using API](#)

2. Use the Import API to import the bot into the Control Room in the destination environment.

[Import files using API](#)

Export files using API

You can export bots with their dependent files using the Export API.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- You must have Export bots, View package, and Check in or Check out permissions to the required folders.
- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- A file ID of the bot you want to export from the public folder.

[List files and folders by workspace API](#)

Note: Users can only view the folders and subfolders they have permissions to access.

- The following API URLs:
 - `https://<your_control_room_url>/v2/blm/export` : To export repository bots
 - `https://<your_control_room_url>/v2/blm/status/{requestId}` : To get export status by request ID
 - `https://<your_control_room_url>/v2/blm/download/{downloadFileId}` : To download the exported bot

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Search for one or more file IDs of the bot you want to export.

3. Use the POST method and endpoint URL: `https://<your_control_room_url>/v2/blm/export`.

The following example request body, exports the bot with `fileId` 2197 along with the packages required for bot.

Request body:

```
{
  "name": "export-docs",
  "fileIds": [
    2197
  ],
  "includePackages": true
}
```

4. Send the request.

The following response body returns the `requestId`.

Response body:

```
{
  "requestId": "987c0de3-b158-4e71-975e-27d10b9a83fb"
}
```

5. Use the GET method and endpoint URL: `<your_control_room_url>/v2/blm/status/{requestId}`

Enter the `requestId` generated in Step 4 to know the status of export.

```
https://192.0.2.0/v2/blm/status/987c0de3-b158-4e71-975e-27d10b9a83fb
```

6. Send the request.

The following response body returns the `status` and `downloadFileId`.

Response body:

```
{
  "requestId": "987c0de3-b158-4e71-975e-27d10b9a83fb",
  "type": "EXPORT",
  "status": "COMPLETED",
  "downloadFileName": "export-docs",
  "downloadFileId": "ZXhwb3J0LWRvY3M=",
}
```

```
"errorMessage": ""
}
```

- Use the GET method and endpoint URL: `<your_control_room_url>/v2/blm/download/{downloadFileId}`

Enter the `downloadFileId` generated in Step 6.

```
https://192.0.2.0/v2/blm/download/ZXhwb3J0LWRvY3M=
```

- Send the request.

The dialog box appears. Browse the path and save the exported package in zip file format.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Next steps

Import the exported file in the private folder of the target Control Room.

Import files using API

You can import bots with their dependent files using the Import API.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- You must have following permissions and licenses:
 - Import bots
 - Manage package
 - Check in permissions to the necessary folders to import bots in the public workspace
 - Bot Creator license to import bots in the private workspace
- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- The following API URLs:
 - `https://<your_control_room_url>/v2/blm/import` : To import repository bot
 - `https://<your_control_room_url>/v2/blm/status/{requestId}` : To get import status by request ID
- To import the Enterprise 11 bots into your Control Room, you must have the required aapkg package that you created using the Bot Lifecycle Management Export API in the Enterprise 11 Control Room instance.

The aapkg package must be present in the same Automation 360 machine where you want to import the Enterprise 11 bots.

You can import the password-protected aapkg packages using the Import API only.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST method and endpoint URL: `https://<your_control_room_url>/v2/blm/import`.

Provide the following parameters for the request header.

- **upload:** Choose the required zip file that you want to import in your Control Room.

Choose the required aapkg file to import Enterprise 11 bots to your Control Room.

- **actionIfExists:** Select either the `SKIP` or `OVERWRITE` option if the file you are importing already exists.
 - **publicWorkspace:** This is a Boolean value. Select either `true` or `false`. Enter `true` if you want to import the file to the public workspace.
3. Send the request.

The following response body returns `requestId`.

Response body:

```
{
  "requestId": "eafef543-2d7a-47f5-81d0-490d09dd68d2"
}
```

4. Use the GET method and endpoint URL: `<your_control_room_url>/v2/blm/status/{requestId}`

Enter the `requestId` generated in Step 3 to know the status of import.

5. Send the request.

The following response body returns the `status`.

Response body:

```
{
  "requestId": "fa4b0c56-fab8-42ef-8d96-fc6b53e1cbaa",
  "type": "IMPORT",
  "status": "COMPLETED",
  "downloadFileName": "",
  "downloadFileId": ""
}
```

```
"errorMessage": ""  
}
```

The **COMPLETED** status indicates that the file is successfully imported. You can find the imported file in your Control Room

Enterprise 11 bots migration:

- The Enterprise 11 bots are imported in the **Bots > My Tasks** folder or **Bots > My Metabots** folder in the .atmx or .mbot file format.
- Use the migration wizard to convert the Enterprise 11 bots files into the .bot format that is supported in Automation 360.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Device pool API

Identify all available device pools or filter device pools by name. Retrieve detailed device pool information for a device by searching for its unique numeric identifier (ID).

A device pool is a logical grouping of devices used by a Bot Runner to distribute and manage the running of unattended bots.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- **List device pools API**

List all available device pools in your Control Room, or filter the list by the name of the device pool.

- **Retrieve details of device pool by ID**

Retrieve the details of a specific device pool by its numeric identifier (ID).

- **Create device pool API**

Device pools are a logical grouping of devices or similar Bot Runner machines on which you can run your workload management automations or scheduled automations. Create a device pool using an API with a unique name and add unattended Bot Runners to the device pool.

Parent topic: [Control Room APIs](#)

List device pools API

List all available device pools in your Control Room, or filter the list by the name of the device pool.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- You must be assigned a role that includes the `View and manage all devices` permission.

This task searches for all device pools that contain the string **finance** in the name.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST and endpoint URL: `<your_control_room_url>/v2/devices/pools/list`.

```
{
  "filter": {
    "operator": "substring",
    "field": "name",
    "value": "finance"
  }
}
```

[Filtering, pagination, and sorting](#)

3. Send the request.

Response body:

```
{
  "page": {
    "offset": 0,
    "total": 15,
    "totalFilter": 1
  },
  "list": [
    {
      "id": "27",
      "name": "finance-device-pool",

```



```
{
  "status": "CONNECTED",
  "detailedStatus": "ALL_DISCONNECTED",
  "automationCount": "0",
  "ownerIds": [
    "48"
  ],
  "deviceCount": "3"
}
```

A successful response lists one or more device pools. Use the ID of a device pool to view the details.

[Retrieve details of device pool by ID](#)

Next steps

If you are performing the steps to run a bot perform this task: [Bot deployment - V3](#)


If you are performing the steps to create an automation schedule perform this task: [Schedule bot to run API](#)

Parent topic: [Device pool API](#)

Retrieve details of device pool by ID

Retrieve the details of a specific device pool by its numeric identifier (ID).

Prerequisites

 **Note:** You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

Ensure you have the following to use this API:

- View and manage ALL device(s): View and manage all the devices, including devices registered by other users.
- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- Device pool id: The unique numeric identifier of the device pool for which you want to retrieve details.

This task searches in the Control Room for a specific device pool. The API passes the device pool {id} as part of the URL string. No request body is required.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.

[Authentication API](#)

2. Use the GET method and endpoint URL: `<your_control_room_url>/v2/devices/pools/{id}`. Enter the device pool ID in the URL.
In this example, the device pool ID is **27**.

```
https://<your_control_room_url>/v2/devices/pools/27
```

3. Send the request.

When the request is successful, all the details associated with the passed device pools ID are returned in response body.

In this example, the name, automation scheme, status, Bot Runners, owners, and consumers, associated with the device pool ID **27** are returned in the response body

Response body:

```
{
  "id": "27",
  "name": "finance-device-pool",
  "description": "Finance department device pool",
  "automationScheme": "ROUND_ROBIN",
  "status": "CONNECTED",
  "timeSlice": "5",
  "timeSliceUnit": "MINUTES",
  "deviceIds": [
    "10",
    "23",
    "41"
  ],
  "ownerIds": [
    "48"
  ],
  "consumerIds": [],
  "detailedStatus": "ALL_DISCONNECTED",
  "updatedBy": "48",
  "updatedOn": "2020-04-27T14:29:05.655896Z",
  "createdBy": "48",
```

```
"createdOn": "2020-04-25T10:46:44.642586Z",  
"tenantUuid": "c0a8f10a-717f-130b-8171-7f4762280000",  
"tenantId": "4"  
}
```

Review the details of the listed device pool to determine if it meets your bot deployment requirements. Some of the fields in the response are used as input to other APIs.

Evaluate these details:

id

The id uniquely identifies this device pool in the Control Room. Use this id as input for the poolIds for bot deployment.

[Bot deployment - V3](#)

name

This is the user-defined name for the device pool. Filter the device pool name using filters in the list device pools task.

[List device pools API](#)

deviceIds

List of the unique numeric IDs for Bot Runner devices that are part of this device pool.

ownerIds

The IDs of the owners of this device pool. Device pool owners can view, edit, or delete the device pool.

consumerIds

The IDs for users who can view this device while running automations.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Parent topic: [Device pool API](#)

Create device pool API

Device pools are a logical grouping of devices or similar Bot Runner machines on which you can run your workload management automations or scheduled automations. Create a device pool using an API with a unique name and add unattended Bot Runners to the device pool.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

Ensure you have the following:

- AAE_Pool_Admin role and View and manage ALL device(s) permission
- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- Endpoint URLs:
 - `<your_control_room_url>/v2/devices/pools`
 - `<your_control_room_url>/v2/devices/list`

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST method and endpoint URL: `<your_control_room_url>/v2/devices/pools`.

If you want to search or get a list of all the available **devicelds**, use the following endpoint URL:

`https://<your_control_room_url>/v2/devices/list`

[Request device details](#)

For example:

```
POST https://192.0.2.0/v2/devices/pools
```

The this example request body enables you to add automation scheme, unattended Bot Runners, owners, and consumers in the device pool.

Request body:

```
{
  "name": "Finance-device-pool",
  "description": "Pool for Finance RPA",
  "deviceIds": [
    "1",
    "10"
  ],
  "automationScheme": "ROUND_ROBIN",
```

```
"ownerIds": [
  "1",
  "24",
  "26"
],
"consumerIds": [
  "21",
  "22"
]
}
```

3. Send the request.

When the request is successful, a unique device pool **id** is returned in the response body. The details of the devices, owners, and consumers associated with the device pool are also provided.

In this example, the response body returns the unique device pool **id** as **4**.

Response body:

```
{
  "id": "4",
  "name": "Finance-device-pool",
  "description": "Pool for Finance RPA",
  "automationScheme": "ROUND_ROBIN",
  "status": "CONNECTED",
  "timeSlice": "5",
  "timeSliceUnit": "MINUTES",
  "deviceIds": [
    "1",
    "10"
  ],
  "ownerIds": [
    "1",
    "24",
    "26"
  ],
  "consumerIds": [
  ],
}
```

```
"detailedStatus": "SOME_CONNECTED",
"updatedBy": "24",
"updatedOn": "2020-05-26T09:26:54.556280800Z",
"createdBy": "24",
"createdOn": "2020-05-26T09:26:54.556280800Z",
"tenantUuid": "4db5b32c-5c4b-4aee-8ca0-f53ec241563c",
"tenantId": "4"
}
```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Next steps

If you are performing the steps to [Create workload automation](#), next [Run bot with queue API](#).

If you are performing the steps to [Deploy a bot](#), next [Bot deployment - V3](#).

Parent topic: [Device pool API](#)

License API

The License API contains endpoints to retrieve Control Room license details (such as expiration date and license mode) and manually sync the Control Room with the license server after license reallocation or renewal.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

Manage licenses

1. [Authenticate the user](#).

Use the POST method to generate an authentication JSON Web Token.

2. [Retrieve Control Room license details API](#).

Retrieve details of the Control Room license, including license type, expiration date, Fail-Safe status, and license server sync status.

3. If the `controlRoomLicenseServerSyncStatus` parameter returns `False`, use the Sync endpoint to update the license allocations and expiration date from the license server.

List the license allocations

1. [Authenticate the user.](#)

Use the POST method to generate an authentication JSON Web Token.

2. **If the license is cloud-based:** If new licenses were purchased, or if licenses were reallocated between Control Room instances, use the Sync endpoint to update the license allocations and expiration date from the license server.
3. [List Control Room licenses](#)

Retrieve Control Room metadata including the license type, number of available licenses, number of licenses used in a specific Control Room instance, and number of licenses used in all Control Room instances.

Retrieve Control Room license details API

Retrieve details of the Control Room license, including license type, expiration date, Fail-Safe status, and license server sync status.

Procedure

1. Use the GET method and endpoint URL: `<your_control_room_url>/v2/license/details`.
2. Send the request.

Response body: In a successful request, this endpoint returns the following data:

- `failSafeStatus`: A numerical value that represents the Fail-Safe status of the Cloud Control Room. Values: 0 means the Control Room is connected, 1 means the Control Room is in Fail-Safe mode, and 2 means the Fail-Safe status has expired.

If the Control Room was configured with a file license, the returned value is always 0.

[Control Room fail-safe status](#)

- `controlRoomLicenseServerSyncStatus`: A Boolean value that determines whether the license allocations in this Control Room are in sync with the license server.
- `licenseMode`: Whether the Control Room license was configured from a file or through a connection with the license server.

[Automation 360 licenses](#)

In this first example response, the `licenseMode` confirms that the Control Room is on a file-based license, thus the `failSafeStatus` is 0 and no value is returned for the `installedCrId`.

```
{
  "type": "PURCHASED",
```

```

    "installationDate": "2020-09-09T15:06:05.211Z",
    "expirationDate": "2021-06-30T18:29:59.999Z",
    "failSafeStatus": 0,
    "controlRoomLicenseServerSyncStatus": true,
    "installedCrId": "",
    "licenseMode": "FileLicense"
  }

```

In the second example response, the `licenseMode` confirms that the Control Room is on a cloud-based license, the `failSafeStatus` has returned 1, which means that requests to other Control Room APIs will fail until the connection with the license server is reestablished, and the `controlRoomLicenseServerSyncStatus` has returned false.

```

{
  "type": "PURCHASED",
  "installationDate": "2020-09-14T18:30:00Z",
  "expirationDate": "2021-04-27T18:30:00Z",
  "failSafeStatus": 1,
  "controlRoomLicenseServerSyncStatus": false,
  "installedCrId": "b96edac7-b7e3-57bf-b857-ad14ac754674",
  "licenseMode": "CloudLicense"
}

```

The REST API responds to each request with an HTTP response code. For details about the response codes, see [API response codes](#).

Next steps

If the `controlRoomLicenseServerSyncStatus` parameter returns `False`, use the Sync endpoint to update the license allocations and expiration date from the license server.

List Control Room licenses

Retrieve Control Room metadata including the license type, number of available licenses, number of licenses used in a specific Control Room instance, and number of licenses used in all Control Room instances.

Request

```
POST https://{ControlRoomURL}
```



```
/v2/license/product/list
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

You can send this API request with or without filter parameters. An API request without any filter parameters specified retrieves the details of all the objects in the Control Room. You can use the filter parameters to retrieve a specific set of file or folder objects instead of fetching all objects in the Control Room. For more information, see [Filtering, pagination, and sorting](#).

Request body without filters:

```
{
  "filter":null,
  "sort":[
    {
      "field":"id",
      "direction":"desc"
    }
  ],
  "page":{
    "offset":0,
    "length":100
  }
}
```

Request body with filters:

```
{
  "filter":{
    "operator":"and",
    "operands":[
      {
        "operator":"lt",
        "field":"purchasedCount",
```

```

        "value": "31"
      },
      {
        "operator": "gt",
        "field": "usedCountByThisCr",
        "value": 3
      }
    ]
  },
  "sort": [
    {
      "field": "id",
      "direction": "asc"
    }
  ],
  "page": {
    "offset": 0,
    "length": 1000
  }
}

```

Request parameters

Parameter	Type	Required	Description
filter	Object	No	<p>Filters the result based on operator, field, or value.</p> <ul style="list-style-type: none"> • operator - Allowed enumerations are NONE, lt, le, eq, ne, ge, gt, substring, and, or, not. • field - Allowed values are name, lastModified, path, or folder. • value - Specify a value for the name, lastModified, path, or folder that you have selected in the field parameter.

Parameter	Type	Required	Description
sort	Array	No	<p>By default, search results are sorted in descending order of their IDs. To specify an alternative sorting, use the sort query parameter.</p> <p>Enter the field by which you want to sort along with the direction asc (ascending) or desc (descending).</p>
page	Object	No	The page object allows you to get the desired pages.

Response

200 OK

```
{
  "page": {
    "offset": 0,
    "total": 12,
    "totalFilter": 1
  },
  "list": [
    {
      "id": 64,
      "name": "ControlRoom",
      "feature": {
        "id": 85,
        "name": "Development",
        "enable": true,
        "purchasedCount": 30,
        "usedCountByThisCr": 5,
        "usedCountByAllCr": 5,
        "availableCount": 25,
      }
    }
  ]
}
```

```

        "licenseCountUnit": "NUMBER"
      },
      "licenseType": "NONE",
      "productMetrics": [

      ]
    }
  ]
}

```

Response Parameters

Parameter	Type	Description
offset	Integer	The starting list offset, used for pagination.
total	Integer	Total number of records.
totalFilter	Integer	Number of records after applying the filter.
List	Array	The array of List object.
List object		
name	String	Returns the product license name, such as <code>ControlRoom</code> , <code>Analytics</code> , <code>BotFarm</code> , <code>Cognitive</code> , <code>DiscoveryBot</code> , and <code>AutomationAnywhereRoboticInterface</code> .
Feature object		
id	Integer	Unique ID of the feature.
name	String	Returns the device license, such as <code>Development</code> for Bot Creator, <code>Runtime</code> for unattended Bot

Parameter	Type	Description
		Runner, and <code>AttendedRuntime</code> for attended Bot Runner licenses.
<code>Enable</code>	Boolean	A flag indicating if a license is enabled or disabled. If the returned value is <code>False</code> , that license cannot be assigned to a user.
<code>PurchasedCount</code>	Integer	The total number of licenses purchased.
<code>usedCountByThisCr</code>	Integer	The number of licenses used in this Control Room instance.
<code>usedCountByAllCr</code>	Integer	The total number of licenses that are used. If the license is file-based, this number matches the number in the <code>usedCountByThisCr</code> parameter. If the license is cloud-based, this parameter returns the total number of licenses used in all of the customer's Control Room instances.
<code>availableCount</code>	Integer	The number of licenses that are available for allocation.
<code>licenseType</code>	String	Shows whether the license is purchased or is a trial license.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Deploy bots using API

Use a combination of endpoints to deploy bots from the public workspace to Bot Runner devices.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

Prerequisites to deploy a bot

Ensure that you are allocated the following resources:

- **View and Run my bots** feature permission
- **Run and schedule** permissions for the folders that contain the bots
- Access to Bot Runner licensed users
- Access to either a default device or a device pool

Note: If the user associated with the Bot Runner license has a default device assigned to their account, the bot deploys on that device. If no default device is assigned, or you want to select a different device, then you must specify a device pool.

Deploy a bot

1. [Authenticate \(username and password\)](#)
2. [List files and folders by workspace API](#)
3. [List available unattended Bot Runners API](#)
4. **Optional:** [List device pools API](#)
5. [Bot deployment - V3](#)
6. [Bot deployment - V4](#)

- **Bot deployment - V3**

As a user with a Bot Runner license, deploy bots on your assigned devices. You can also pass variables to bots when they are deployed.

Parent topic: [Control Room APIs](#)

Bot deployment - V3

As a user with a Bot Runner license, deploy bots on your assigned devices. You can also pass variables to bots when they are deployed.

Request

```
POST https://{ControlRoomURL}/v3/automations/deploy
```

Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Note: Bot deploy request can be made without any input fields. If you specify inputs, make sure that the bot to successfully receive these mapped in values, the variable for that bot must be marked as input. Additionally, the name of the variable in the bot has to match the value that is being mapped in the request body.

Request body with LIST input:

```
{
  "fileId": 86,
  "runAsUserIds": [
    3
  ],
  "poolIds": [],
  "overrideDefaultDevice": false,
  "callbackInfo": {
    "url": "https://callbackserver.com/storeBotExecutionStatus",
    "headers": {
      "X-Authorization": "{{token}}"
    }
  },
  "botInput": {
    "iTestList": {
      "type": "LIST", //Type can be [ STRING, NUMBER, BOOLEAN, LIST, DICTIONARY, DATETIME ]
      "list": [
        { "type":"STRING",
          "string": "TestValues1"
        },
        { "type":"STRING",
          "string": "TestValues2"
        }
      ]
    }
  }
}
```

```

    ] //key must match type, in this case string
  }
}
}

```

Request body with STRING input:

```

{
  "fileId": 87,
  "runAsUserIds": [
    3
  ],
  "poolIds": [],
  "overrideDefaultDevice": false,
  "callbackInfo": {
    "url": "https://eogplyk2wlo3ec2.m.pipedream.net",
    "headers": {
      "X-Authorization": "{{token}}"
    }
  },
  "botInput": {
    "sInput1": {
      "type": "STRING",
      "string": "Test Values1"
    },
    "sInput2": {
      "type": "STRING",
      "string": "Test Values2"
    }
  }
}

```

Request body with NUMBER input:

```

{
  "fileId": 87,
  "runAsUserIds": [

```



```

    3
  ],
  "poolIds": [],
  "overrideDefaultDevice": false,
  "callbackInfo": {
    "url": "https://eogplyk2wlo3ec2.m.pipedream.net",
    "headers": {
      "X-Authorization": "{{token}}"
    }
  },
  "botInput": {
    "sInput1": {
      "type": "NUMBER",
      "number": 123
    },
    "sInput2": {
      "type": "NUMBER",
      "number": 345
    }
  }
}

```

Request body with DICTIONARY input:

```

{
  "fileId": 86,
  "runAsUserIds": [
    3
  ],
  "botInput": {
    "iTestList": {
      "type": "DICTIONARY", //Type can be [ STRING, NUMBER, BOOLEAN, LIST, DIC
TIONARY, DATETIME ]
      "dictionary": [
        {
          "key": "key1",
          "value": {

```

```

        "type": "STRING",
        "string": "value1"
    },
    {
        "key": "key2",
        "value": {
            "type": "STRING",
            "string": "value2"
        }
    }
] //key must match type, in this case string
}
}
}

```


Request body with DATE TIME input:

```

{
  "fileId": 87,
  "runAsUserIds": [
    3
  ],
  "botInput": {
    "dt_input1": {
      "type": "DATETIME",
      "string": "2022-04-07T00:15:00-06:00[USA/New York]"
    },
    "dt_input2": {
      "type": "DATETIME",
      "string": "2022-04-07T00:15:05-06:00[USA/New York]"
    }
  }
}

```

Request Parameters

Parameter	Type	Required	Description
fileId	Integer	Yes	File Id of bot to be deployed. List files and folders by workspace API
automationName	String	No	Name of the automation to be deployed.
runAsUserIds	Integer	Yes	List of runAs user ids to deploy bot. The bot will be deployed on associated default device for each <code>runAsUserIds</code> or on one of the devices from the device pool(s), if provided. List available unattended Bot Runners API
callbackInfo	String	No	<p><code>callbackInfo</code> provides the callback API URL (For example, <code>https://callbackserver.com/storeBotExecutionStatus</code>) and authentication token for the callback server. After the bot is deployed, the Control Room sends the deployment status and output variable values to this callback server. For example: To test the call back you can create an account in Pipedream and use the endpoint (similar to <code>https://eogp1yk2w1o3ec2.m.pipedream.net</code>) to receive the status and the output variable values.</p> <div>  Note: The callback server must accept POST calls to receive the bot execution data and the </div>

Parameter	Type	Required	Description
			deployment status from the Control Room.
poolIds	Integer	No	<p>You will define the <code>poolIds</code> only when you are running it against a device pool or a pool of runners instead of an individual runner. Identifier of a device pool that has at least one active device. If you enter multiple pool IDs, enter the values in the order of which you want the API to check for available devices. If none of the devices are available at the time of deployment, the automation is queued.</p> <div> <p>Note: If the user associated with the Bot Runner license has a default device assigned to their account, the bot deploys on that device. If no default device is assigned, or you want to select a different device, then you must specify a device pool.</p> <p>List device pools API</p> </div>
overrideDefaultDevice	Boolean	No	<p>If the Bot Runner user is assigned to a default device and you want to specify a device pool, set this parameter to <code>true</code>.</p> <p>If deploying to the default device, set this parameter to <code>false</code>.</p>

Parameter	Type	Required	Description
runElevated	Array	No	Whether to deploy the bot using elevated permissions or not. Possible values include - <code>false</code> , <code>true</code> .
numOfRunAsUsersToUse	Integer	No	<p>Specifies how many Bot Runners to use from the list of runAsUserIds provided. A weighted system algorithm selects the Bot Runners with the least number of queued tasks.</p> <ul style="list-style-type: none"> System will pick the specified number of <code>runAsUsers</code> with the least number of tasks queued for the user at the moment of deploy request. If 0 then all the users will be used. If the number is greater than the number of <code>runAsUsers</code> provided or less than 0 it will error out.
automationPriority	String	No	The automation Priority. By default it is set to <code>PRIORITY_MEDIUM</code> . Possible values for <code>automationPriority</code> includes: <code>PRIORITY_MEDIUM</code> , <code>PRIORITY_HIGH</code> , and <code>PRIORITY_LOW</code> .
botLabel	String	No	Label for the bot to be deployed.
botInput	Object	No	A data structure containing a botInput Object. See below for more details.
botInput Object			
type*	Any	No	By default it is <code>STRING</code> . Possible values for <code>type</code> includes: <code>STRING</code> , <code>NUMBER</code> , <code>BOOLEAN</code> , <code>FILE</code> , <code>ITERATOR</code> , <code>LIST</code> , <code>DICTIONARY</code> , <code>TABLE</code> , <code>VARIABLE</code> ,

Parameter	Type	Required	Description
			CONDITIONAL , WINDOW , TASKBOT , DATETIME , UIOBJECT , RECORD , EXCEPTION , CREDENTIAL , COORDINATE , IMAGE , REGION , PROPERTIES , TRIGGER , CONDITIONALGROUP , FORM , FORMELEMENT , HOTKEY , and WORKITEM .
*The structure of the input varies depending on the type you want to input. This topic provides you with a few of the mostly used sample.			

Response

200 OK

For more information on the return codes, see [API response codes](#).

```
{
  "deploymentId": "340a2949-aa44-41ab-af9b-f9343ae2581c",
  "automationName": "Sample-bot-deploy"
}
```

Tip: Check the bot deployment status and the bot output variables using [Activity list](#).

Response Parameters

Parameter	Type	Description
deploymentId	String	The deployment Id created.

Parameter	Type	Description
automationName	String	Name of the deployed automation. If this name is not supplied in the request, a random name is assigned to the deployed automation.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Workload Management API

Use the Workload Management API to programmatically manage and create Work Item models, queues, Work Items, and automations in your Control Room.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

Prerequisites

To create a automation process, ensure that you are allocated the following roles and permissions:

- AAE_Pool Admin role and View all devices permission
- AAE_Queue Admin role or Create queues permission
- Run bot permission
- Run or schedule permission on the bots folder
- Device pool consumer permission

Create workload automation

1. [Authenticate the user](#).

Use the POST method to generate an authentication JSON Web Token.

2. Create workload management queues:
 - a. [Create Work Item model API](#)
 - b. [Create queues API](#)
 - c. [Add queue owner or member API](#)

- d. [Add queue participants API](#)
- e. [Add queue consumer API](#)
- f. [Add Work Items to the queue API](#)
3. [Create device pool API](#)
4. [Run bot with queue API](#)

Retrieve workload management entities using list APIs

When you create workload management queues for workload automation (see previous section), you can use the Workload Management list APIs to retrieve a list of workload management entities such as Work Item models, queues, and Work Items in queues associated with the Control Room.

[Workload Management list APIs](#)

Mapping of UI to API status for workload management work item processing

The following table shows the UI status and its corresponding API status.

UI	API
New	NEW
Ready to Run	READY_TO_RUN
Active	ACTIVE
Completed	COMPLETED
Failed	FAILED
On Hold	ON_HOLD
Data error	DATA_ERROR

Create Work Item model API

Define a Work Item structure (model) for processing in a queue. This enables you to manually upload the Work Item from the system in the absence of ready data in a file.

Prerequisites

You must have the following:

- Create queues or AAE_Queue Admin permission
- The endpoint URL: `<your_control_room_url>/v3/wlm/workitemmodels`

Procedure

1. Use the POST method and endpoint URL: `<your_control_room_url>/v3/wlm/workitemmodels`.

Enter the parameters such as Work Item model **name**, **attribute** names, and **type** value in the request body to create a Work Item structure. The value of **type** can be `TEXT`, `NUMBER`, or `DATE`, depending on the **attribute** value format.

For example:

```
POST https://192.0.2.0/v3/wlm/workitemmodels
```

In this example, the Work Item model **name** is `Finance-template` and it includes the `first_name`, `last_name`, and `email` as **attributes**. For these attributes, use `TEXT` value as the **type** parameter.

Request body:

```
{
  "name": "Finance-template",
  "attributes": [
    {
      "name": "first_name",
      "type": "TEXT"
    },
    {
      "name": "last_name",
      "type": "TEXT"
    },
    {
      "name": "email",
      "type": "TEXT"
    }
  ]
}
```

```
]
}
```

2. Send the request.

When the request is successful, the Work Item model **id** and the display column **id** are returned in the response. You will use these IDs when you create queues.

In this example, the response body returns the Work Item model **id** as **10** and the display column **id** as **59**, **60**, **61** for the **first_name**, **last_name**, and **email**, respectively.

Response body:

```
{
  "id": 10,
  "createdBy": 24,
  "createdOn": "2020-05-26T06:14:27.520336200Z",
  "updatedBy": 24,
  "updatedOn": "2020-05-26T06:14:27.520336200Z",
  "tenantId": 1,
  "version": 0,
  "tenantUuid": "4db5b32c-5c4b-4aee-8ca0-f53ec241563c",
  "name": "fin",
  "attributes": [
    {
      "id": 59,
      "name": "first_name",
      "type": "TEXT"
    },
    {
      "id": 60,
      "name": "last_name",
      "type": "TEXT"
    },
    {
      "id": 61,
      "name": "email",
      "type": "TEXT"
    }
  ]
}
```

```
]
}
```

The REST API responds to each request with an HTTP response code. For details about the response codes, see [API response codes](#).

Next steps

[Create queues](#)

Create queues API

A queue is one of the main building blocks of workload management. It holds specific sets of data that your bot is expecting for automation.

Prerequisites

Ensure you have the following:

- Create queues or AAE_Queue Admin permission
- The endpoint URLs:
 - `<your_control_room_url>/v3/wlm/queues`
 - `<your_control_room_url>/v3/wlm/workitemmodels/list`

Procedure

1. Use the POST method and endpoint URL: `<your_control_room_url>/v3/wlm/queues` . Enter values for the following parameters to create a queue. Use the same **workItemModelId** and **displayColumnIds** that you received in response when you created the Work Item model.

Note: If you want to search or get a list of all the available **workItemModelId**, use the endpoint URL `<your_control_room_url>/v3/wlm/workitemmodels/list` .

[List workload management queues](#)

For example:

```
POST https://192.0.2.0/v3/wlm/queues
```

In this example, use **workItemModelId** as `10` and **displayColumnIds** as `59` , `60` , `61` .

Request body:

```
{
  "name": "Finance-Q",
  "description": "Queue for Finance",
  "reactivationThreshold": 1,
  "displayColumns": [
    "first_name",
    "last_name",
    "email"
  ],
  "workItemProcessingOrders": [

  ],
  "workItemModelId": 10,
  "displayColumnIds": [
    59,
    60,
    61
  ]
}
```

2. Send the request.

When the request is successful, a queue **id** is returned in the response body. This queue **id** will be used in the subsequent tasks when you add owners, participants, consumers, and Work Items in the queue. In this example, the response body returns the queue **id** as **17**.

Response body:

```
{
  "id": "17",
  "createdBy": "24",
  "createdOn": "2020-05-26T06:13:57.644499300Z",
  "updatedBy": "24",
  "updatedOn": "2020-05-26T06:13:57.644499300Z",
  "tenantId": "1",
  "version": "0",
  "tenantUuid": "4db5b56c-5c2b-4aee-8ca0-f53ec241563c",
  "name": "Finance-Q",
  "description": "Queue for Finance",
```

```

"reactivationThreshold": "1",
"status": "NOT_IN_USE",
"manualProcessingTime": "0",
"manualProcessingTimeUnit": "",
"workItemProcessingOrders": [

],
"workItemModelId": "10",
"displayColumnIds": [
    "59",
    "60",
    "61"
],
"considerReactivationThreshold": false
}

```

The REST API responds to each request with an HTTP response code. For details about the response codes, see [API response codes](#).

Next steps

[Add queue owner or member](#)

Add queue owner or member API

Add queue owners who can create, edit, and view queues. The queue creator is the default queue owner and can add other Control Room users as queue owners if required.

Prerequisites

You must have the following:

- Create queues or AAE_Queue Admin permission
- The endpoint URLs:
 - `<your_control_room_url>/v3/wlm/queues/{queueId}/members/{userId}`
 - `<your_control_room_url>/v1/usermanagement/users/list`
 - `<your_control_room_url>/v3/wlm/queues/list`

Procedure

1. Use the PUT method and endpoint URL: `<your_control_room_url>/v3/wlm/queues/{queueId}/members/{userId}`.

Enter the **queueId** to which you want to add the owner and the **userId** that will be the owner of the queue. This **queueId** is the same ID that was returned when you created the queue.

Note:

- If you want to search or get a list of all the available **queueId**, use the endpoint URL `<your_control_room_url>/v3/wlm/queues/list`.

[List workload management queues](#)

- If you want to search or get a list of all the **userId**, use the endpoint URL: `<your_control_room_url>/v1/usermanagement/users/list`.

[Search for users API](#)

For example, use **queueId** as `17` and **userId** as `1`.

```
PUT https://192.0.2.0/v3/wlm/queues/17/members/1
```

Add one or more **permissions** in the request body to allow the user to perform the specific queue actions. In this example, `manage` and `own` permissions are added.

Request body:

```
{
  "permissions": [
    "manage",
    "own"
  ]
}
```

2. Send the request.

When the request is successful, the required user is added as a queue owner.

In this example, the user with **userId** as `1` is added as a queue owner.

Response body:

```
{
  "id": 1,
  "permissions": [
    "own",
```

```
"manage"
]
}
```

The REST API responds to each request with an HTTP response code. For details about the response codes, see [API response codes](#).

Next steps

[Add queue participants](#)

Add queue participants API

Add queue participants from different roles defined in the Control Room.

Prerequisites

You must have the following:

- Create queues or AAE_Queue Admin permission
- The endpoint URLs:
 - `<your_control_room_url>/v3/wlm/queues/{queueId}/participants`
 - `<your_control_room_url>/v3/wlm/queues/list`

Procedure

1. Use the POST method and endpoint URL: `<your_control_room_url>/v3/wlm/queues/{queueId}/participants`.

Enter the **queueId** to which you want to add the participants. This **queueId** is the same ID that was returned when you created the queue.

Note: If you want to search or get a list of all the available **queueId**, use the endpoint URL `<your_control_room_url>/v3/wlm/queues/list`.

[List workload management queues](#)

For example, use **queueId** as `17`.

```
POST https://192.0.2.0/v3/wlm/queues/17/participants
```

Enter one or more role **id** in the request body that you want to add as queue participants. In this example, one role **id** as `21` is added as queue participant.

Request body:

```
[
  {
    "id": 21
  }
]
```

2. Send the request.

When the request is successful, the participants are added to the queue.

In this example, the participant with role **id** as 21 is added to the queue.

Response body:

```
[
  {
    "id": 21,
    "createdBy": null,
    "createdOn": null,
    "updatedBy": null,
    "updatedOn": null,
    "tenantId": null,
    "version": 0,
    "tenantUuid": null,
    "description": null,
    "name": null,
    "accessRestriction": null,
    "permissions": [

    ],
    "countPrincipals": 0,
    "principals": [

    ]
  }
]
```


The REST API responds to each request with an HTTP response code. For details about the response codes, see [API response codes](#).

Next steps

[Add queue consumer](#)

Add queue consumer API

Add queue consumers from different roles defined in the Control Room.

Prerequisites


You must have the following:

- Create queues or AAE_Queue Admin permission
- The endpoint URLs:
 - `<your_control_room_url>/v3/wlm/queues/{queueId}/consumers`
 - `<your_control_room_url>/v3/wlm/queues/list`

Procedure

1. Use the POST method and endpoint URL: `<your_control_room_url>/v3/wlm/queues/{queueId}/consumers`.

Enter the **queueId** to which you want to add the consumers. This **queueId** is the same ID that was returned when you created the queue.

 **Note:** If you want to search or get a list of all the available **queueId**, use the endpoint URL `<your_control_room_url>/v3/wlm/queues/list`.

[List workload management queues](#)

For example, use **queueId** as `17`.

```
POST https://192.0.2.0/v3/wlm/queues/17/consumers
```

Enter one or more role **id** in the request body that you want to add as queue consumers. In this example, one role **id** as `21` is added as queue consumer.

Request body:

```
[
  {
    "id": 21
  }
]
```

2. Send the request.

When the request is successful, the consumers are added to the queue.

In this example, the consumer with role **id** as 21 is added to the queue.

Response body:

```
[
  {
    "id": 21,
    "createdBy": null,
    "createdOn": null,
    "updatedBy": null,
    "updatedOn": null,
    "tenantId": null,
    "version": 0,
    "tenantUuid": null,
    "description": null,
    "name": null,
    "accessRestriction": null,
    "permissions": [

    ],
    "countPrincipals": 0,
    "principals": [

    ]
  }
]
```

The REST API responds to each request with an HTTP response code. For details about the response codes, see [API response codes](#).

Next steps

[Add Work Items to the queue](#)

Add Work Items to the queue API

Add or insert Work Items to an existing queue in the Control Room per the defined model or structure.

Prerequisites

- The user must be a Queue owner and participant
- You must have the endpoint URLs:
 - `<your_control_room_url>/v3/wlm/queues/{queueId}/workitems`
 - `<your_control_room_url>/v3/wlm/queues/list`

Procedure

1. Use the POST method and endpoint URL: `<your_control_room_url>/v3/wlm/queues/{queueId}/workitems`.

Enter the **queueId** to which you want to add a Work Item in the queue.

Note: If you want to search or get a list of all the available **queueId**, use the endpoint URL `<your_control_room_url>/v3/wlm/queues/list`.

[List workload management queues](#)

For example, use **queueId** as `17`.

```
POST https://192.0.2.0/v3/wlm/queues/17/workitems
```

Enter a **workitem** in the request body.

Note: In the request body, ensure there are no hidden, invalid, new line, or EOL characters. You can check for these using text or source code editors.

Request body:

```
{
  "workItems": [
    {
```

```
    "json": {
      "first_name": "Jane",
      "last_name": "Doe",
      "email": "jane.doe@example.com"
    }
  }
]
```

2. Send the request.

When the request is successful, a unique Work Item **id** is returned in the response body and the Work Items are added to the queue per the defined Work Item model or structure.

In this example, the Work Item with the **first_name** as **Jane**, **last_name** as **Doe**, and **email** as **jane.doe@example.com** is added to the queue based on the defined structure.

Response body: (truncated output)

```
{
  "id": 77,
  "createdBy": 24,
  "createdOn": "2020-05-19T17:41:57.602092100Z",
  "updatedby": 24,
  "updatedOn": "2020-05-26T09:13:31.090241700Z",
  "version": 2,
  "json": {
    "first_name": "Jane",
    "last_name": "Doe",
    "email": "jane.doe@example.com"
  },
  "result": "",
  "deviceId": 0,
  "status": "NEW",
  "col1": "1.0",
  "col2": "",
  ...,
  "co21": "",
  "deviceUserId": 0,
  "queueId": 5,
```

```
"comment": "",
"automationId": 0,
"totalPausedTime": 0,
"error": ""
}
```

3. **Optional:** If you want to add multiple Work Items, call the API using a list of Work Item JSON objects.

Request body:

```
{
  "workItems": [
    {
      "json": {
        "DATA": "mydata",
        "TRN_ID": "A11"
      }
    },
    {
      "json": {
        "DATA": "mydata",
        "TRN_ID": "A11"
      }
    }
  ]
}
```

Response body:

```
{
  "list": [
    {
      "id": "40957",
      "createdBy": "25",
      "createdOn": "2021-11-24T01:53:10.175335900Z",
      "updatedBy": "25",
      "updatedOn": "2021-11-24T01:53:10.175335900Z",
      "version": "0",

```

```
"json": {
  "TRN_ID": "A11",
  "DATA": "mydata"
},
"result": "",
"deviceId": "0",
"status": "NEW",
"col1": "A11",
"col2": "",
"col3": "",
"col4": "",
"col5": "",
"deviceUserId": "0",
"queueId": "0",
"comment": "",
"automationId": "0",
"totalPausedTime": "0",
"error": "",
"col6": "",
"col7": "",
"col8": "",
"col9": "",
"col10": "",
"jobExecutionId": ""
},
{
  "id": "40958",
  "createdBy": "25",
  "createdOn": "2021-11-24T01:53:10.198337200Z",
  "updatedBy": "25",
  "updatedOn": "2021-11-24T01:53:10.198337200Z",
  "version": "0",
  "json": {
    "TRN_ID": "A11",
    "DATA": "mydata"
  },

```

```

        "result": "",
        "deviceId": "0",
        "status": "NEW",
        "col1": "A11",
        "col2": "",
        "col3": "",
        "col4": "",
        "col5": "",
        "deviceUserId": "0",
        "queueId": "0",
        "comment": "",
        "automationId": "0",
        "totalPausedTime": "0",
        "error": "",
        "col6": "",
        "col7": "",
        "col8": "",
        "col9": "",
        "col10": "",
        "jobExecutionId": ""
    }
]
}

```

4. **Optional:** If you want to update the Work Item data, when the automation is running, you need to perform the following steps:
 - a. Pause the automation. Use the PUT method and the following endpoint URL:
`<your_control_room_url>/v3/wlm/automations/{id}`
 - b. Update the Work Item by using queue ID and Work Item ID. Use the PUT method and the following endpoint URL: `<your_control_room_url>/v3/wlm/queues/{id}/workitems/{workitemId}`
 - c. Resume the automation. Use the PUT method and the following endpoint URL:
`<your_control_room_url>/v3/wlm/automations/{id}`

Next steps

[Create an automation to run a bot with a queue](#)

Run bot with queue API

Create an automation to collectively process all the Work Items of a queue across all the Bot Runners present in one or more device pools using the API.

Prerequisites

- You must have the following permissions:
 - Run bot
 - Run or schedule permission on the bot folder
 - Queue consumer
 - Device pool consumer
- You must have the endpoint URLs:
 - `<your_control_room_url>/v3/wlm/automations`
 - `<your_control_room_url>/v1/usermanagement/users/list`
 - `<your_control_room_url>/v3/wlm/queues/list`
 - `<your_control_room_url>/v2/devices/pools/list`

Procedure

- Use the POST method and endpoint URL: `<your_control_room_url>/v3/wlm/automations`.

Enter parameters such as **name** of the automation, **fileId**, **filename**, **queueId**, **queueName**, **runAsUserIds**, and **poolId**.

- If you want to search or get a list of all the available **queueId**, use the endpoint URL `<your_control_room_url>/v3/wlm/queues/list`.

[List workload management queues](#)

- If you want to search or get a list of all the **runAsUserIds**, use the endpoint URL: `<your_control_room_url>/v1/usermanagement/users/list`.

[Search for users API](#)

- If you want to search or get a list of all the **poolId**, use the endpoint URL: `<your_control_room_url>/v2/devices/pools/list`.

[List device pools API](#)

In this example, the parameters are entered as follows:

- Automation **name** as `Finance-RPA-Run`
- Bot **fileName** as `wlmql`
- runAsUserIds** as `4` and `5` that will log in to the device to run the automation
- queueId** as `17`, associated with the queue to run the automation
- poolId** as `1` that is associated with the pool

Request body:

```
{
  "name": "Finance-RPA-Run",
  "automationName": "Finance-RPA-Run",
  "fileName": "wlmq1",
  "botInput": {

  },
  "status": "ACTIVE",
  "description": "WLM for Finance",
  "rdpEnabled": false,
  "setAsDefaultDevice": false,
  "poolIds": [

  ],
  "workspaceName": "public",
  "timeZone": "Asia/Calcutta",
  "runAsUserIds": [
    "4",
    "5"
  ],
  "queueId": "17",
  "poolId": "1"
}
```

2. Send the request.

When the request is successful, a unique automation **id** is returned in the response body after the workload management automation run successfully. The details of the associated queue name and ID, and the user name IDs for which the automation is run are also provided.

In this example, the response body returns the unique automation **id** as **12**.

Response body:

```
{
  "id": "12",
  "name": "Finance-RPA-Run",
  "status": "ACTIVE",
```

```
"description": "WLM for Finance",
"rdpEnabled": false,
"priority": "1",
"queueId": "17",
"queueName": "Finance-Q",
"poolId": "1",
"runAsUserIds": [
  "4",
  "5"
],
"fileId": "17",
"startedOn": "2020-05-26T09:42:51.958893800Z",
"startedBy": "24",
"createdBy": "24",
"createdOn": "2020-05-26T09:42:51.958893800Z",
"updatedBy": "24",
"updatedOn": "2020-05-26T09:42:51.958893800Z",
"tenantId": "1",
"version": "0",
"tenantUuid": "4db5b32c-5c4b-4aee-8ca0-f53ec241563c"
}
```

The REST API responds to each request with an HTTP response code. For details about the response codes, see [API response codes](#).

Workload Management list APIs

List APIs use the POST method to return the list of the entities. You can use the Workload Management list APIs and get the list of workload management entities, for example, Work Item models, queues, and Work Items in the queues.

Use the following Workload Management APIs to get the list of the workload management entities:

- [List Work Item models](#)
- [List workload management queues](#)
- [List Work Items in queue](#)

List Work Item models

Use the Work Item model list API to get the list of all the Work Item templates that are associated with the specified Control Room.

Prerequisites

You must Queue Consumer permission.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST method and endpoint URL: `<your_control_room_url>/v3/wlm/workitemmodels/list`.

Use the following URL and leave the request body blank to request information about all available Work Item models:

```
https://192.0.2.0/v3/wlm/workitemmodels/list
```

3. Send the request.

Because there is no filtering used in the request, a successful response returns all of the queues for the specified Control Room. In this example, the response returned data for two Work Item models.

Response body:

```
{
  "page": {
    "offset": 0,
    "total": 50,
    "totalFilter": 50
  },
  "list": [
    {
      "id": "1",
      "name": "q1",
      "attributes": [
        {
          "name": "Customer Name",
          "type": "TEXT",
          "id": "2"
```

```

    },
    {
      "name": "email",
      "type": "TEXT",
      "id": "4"
    },
  ],
},
{
  "id": "2",
  "name": "q2",
  "attributes": [
    {
      "name": "Customer Name",
      "type": "TEXT",
      "id": "7"
    },
    {
      "name": "email",
      "type": "TEXT",
      "id": "9"
    },
  ],
}
]
}

```

List workload management queues

Use the Workload Management queues list API to get the list of all the queues that are associated with the specified Control Room. To execute this API, you must have Queue Consumer permission.

Request

```
POST http://{{ControlRoomURL}}/v3/wlm/queues/list
```

Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request body without filters:

```
{
  "sort": [
    {
      "field": "name",
      "direction": "asc"
    }
  ],
  "filter": {
  },
  "page": {
    "offset": 0,
    "total": 100,
    "totalFilter": 100,
    "length": 200
  }
}
```

Request body with filters:

```
{
  "sort": [
    {
      "field": "name", //set sort by criteria
      "direction": "asc"
    }
  ],
  "filter": {
    "operator": "and",
    "operands": [
```

```

    {
      "operator": "eq",
      "value": "DRAFT",
      "field": "status"
    },
    {
      "operator": "substring",
      "value": "Test",
      "field": "name"
    }
  ]
},
"fields": [],
"page": { //return a limited number of results with offset for pagination
  "offset": 0,
  "total": 100,
  "totalFilter": 100,
  "length": 200
}
}

```

Request Parameters

Parameter	Type	Required	Description
sort	Array	No	<p>By default, search results are sorted in descending order with respect to their ids. An alternative sorting is specified using the sort query parameter.</p> <p>Enter the field by which you want to sort along with the direction <code>asc</code> (ascending) or <code>desc</code> (descending). For more information on sorting, see Filtering, pagination, and sorting</p>
filter	Object	No	<p>Filters the result. For more information on sorting, see Filtering, pagination, and sorting</p>

Parameter	Type	Required	Description
fields	Array	No	Filter the result based on the fields.
page	Object	No	The page object allows you to get the desired pages. For more information on pagination rules, see Filtering, pagination, and sorting

For the above sample request, the response returns all the details of the queue that has **Test** in their name and **NOT_IN_USE** as status.

Response

```
{
  "page": {
    "offset": 0,
    "total": 17,
    "totalFilter": 1
  },
  "list": [
    {
      "id": "32",
      "createdBy": "873",
      "createdOn": "2022-03-02T05:51:53.855636Z",
      "updatedBy": "873",
      "updatedOn": "2022-03-02T05:54:18.745722Z",
      "tenantId": "1",
      "version": "1",
      "tenantUuid": "b6e4eb84-f7ef-4dfd-a432-725b71de8142",
      "name": "for-docs-test",
      "description": "",
      "reactivationThreshold": "1",
      "status": "NOT_IN_USE",
      "manualProcessingTime": "0",
      "manualProcessingTimeUnit": "SECONDS",
      "workItemProcessingOrders": [],
    }
  ]
}
```

```

        "workItemModelId": "28",
        "displayColumnIds": [
            "156",
            "157",
            "158",
            "159",
            "160"
        ],
        "considerReactivationThreshold": false
    }
]
}

```

Response Parameters

Parameter	Type	Description
offset	Integer	The starting list offset, used for pagination.
total	Integer	Total number of records.
totalFilter	Integer	Number of records after applying the filter.
List	Array	The array of List of queues.
List roles object		
id	Integer	The unique Id of a specific queue.
createdBy	Integer	Id of the user who created the role.
createdOn	String	The creation timestamp of the role.
updatedBy	Integer	Id of the user who made a latest update to the role.

Parameter	Type	Description
updatedAt	String	The latest update timestamp of the role.
tenantId	Integer	Id of the tenant.
version	Integer	Version of the role instance.
tenantUuid	String	Unique Id of the tenant.
name	String	Name of queue.
description	String	Description of queue.
reactivationThreshold	Integer	Minimum number of workItems in the queue to reactivate queue.
status	String	The current status of the queue. The values can be DRAFT, IN_USE, or NOT_IN_USE.
manualProcessingTime	Integer	Efforts required to process the queue manually.
manualProcessingTimeUnit	String	Unit of the manual Processing time. The values can be SECONDS, MINUTES, HOURS, or DAYS.
workItemProcessingOrders	Array	An array of work item processing orders.
workItemModelId	Integer	Id of the Work Item Model.
displayColumnIds	Array	An array displaying the list of column Ids.
considerReactivationThreshold	Boolean	Flag to indicate if the Reactivation Threshold is required or not.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

List Work Items in queue

Use the Workload Management Work Item list API to get the list of all the Work Items in the queues that are associated with the specified Control Room.

Prerequisites

You must have the following:

- Queue Consumer permission
- All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
- The endpoint URL: `<your_control_room_url>/v3/wlm/queues/{queueId}/workitems/list`

Leave the request body blank to request information on all available Work Items. Add one or more filter parameters in the request body to limit the information returned from all available workload management Work Items.

Supported filterable parameters:

status

The status of queue for example: New, On hold, Failed, Completed, Data error, Active, and Ready to run.

- **Field:** status
- **Type:** string

```
{
  "filter": {
    "operator": "eq",
    "value": "ACTIVE",
    "field": "status"
  }
}
```

result

The Work Item result string. For example, the Work Item was completed or skipped.

- **Field:** result
- **Type:** string

```
{
  "filter": {
    "operator": "substring",
    "value": "skipped",
    "field": "result"
  }
}
```

col

The column number corresponding to the custom column name. For example, email, firstname, and lastname.

- **Field:** col
- **Type:** string

```
{
  "filter": {
    "operator": "substring",
    "value": "Brian",
    "field": "col1"
  }
}
```

Procedure

1. Use the POST method to generate an authentication JSON Web Token.
[Authentication API](#)
2. Use the POST method and endpoint URL: `<your_control_room_url>/v3/wlm/queues/{queueId}/workitems/list`

For example, enter the `queueId` as `20` in the following URL for which you want to get the Work Items:

```
https://192.0.2.0/v3/wlm/queues/20/workitems/list
```

Use filters in the request body to retrieve the list of all the Work Items that are in **NEW** status and have **Brian** in their first_name (**coll**).

Request body:

```
{
  "sort": [
    {
      "field": "computedStatus",
      "direction": "asc"
    }
  ],
  "filter": {
    "operator": "and",
    "operands": [
      {
        "operator": "eq",
        "value": "NEW",
        "field": "status"
      },
      {
        "operator": "substring",
        "value": "Brian",
        "field": "coll"
      }
    ]
  },
  "page": {
    "offset": 0,
    "total": 5,
    "totalFilter": 1,
    "length": 100
  }
}
```

3. Send the request.

- In the REST Client, click **SEND**.

- In the Swagger interface, click **Execute**.

The response returns all the details of the Work Item that has **Brian** in their first_name (**col1**) and status is **NEW**.

Response body:

```
{
  "page": {
    "offset": 0,
    "total": 5,
    "totalFilter": 1
  },
  "list": [
    {
      "id": "11804",
      "createdBy": "24",
      "createdOn": "2020-05-26T10:19:34.786711300Z",
      "updatedBy": "24",
      "updatedOn": "2020-05-26T10:19:34.786711300Z",
      "version": "1",
      "json": {},
      "result": "",
      "deviceId": "0",
      "status": "NEW",
      "col1": "Brian",
      "col2": "Matthews",
      "col3": "bmatthews0@example.com",
      "col4": "",
      "col5": "",
      "deviceUserId": "0",
      "queueId": "20",
      "comment": "",
      "automationId": "0",
      "totalPausedTime": "0",
      "error": "",
      "col6": "",
      "col7": "",
      "col8": "",
    }
  ]
}
```

```

    "col9": "",
    "col10": ""
  }
]
}

```

Migration APIs

Use migration APIs to migrate MetaBots and TaskBots that were created in Enterprise client versions Enterprise 11 and Enterprise 10 to Automation 360. Use this page to review the migration prerequisites and access Enterprise 11 and Enterprise 10 Migration APIs.

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

Prerequisites

Complete the prerequisites before migrating Enterprise 11 bots to Automation 360. [Prerequisite tasks for migrating bots](#)

Enterprise 11 Migration APIs

1. [Start migration API](#)
2. [Migrate all bots in a sub-folder API](#)
3. [List migration results API](#)
4. [Bot migration results by id API](#)
5. [Migration action mapping results API](#)

Enterprise 10 Migration APIs

1. [Connect to Enterprise 10 database](#)
2. [Validate master key and repository path](#)
3. [Initiate Enterprise 10 data migration process](#)

(Optional) Use any the following APIs to retrieve the migration results.

- [Retrieve migrated roles](#)
- [Retrieve migrated users](#)
- [Retrieve migrated credentials](#)
- [Retrieve migrated bots](#)
- [Retrieve migrated schedules](#)

Start migration API

Use this API to migrate bots (TaskBots and MetaBots) created using the Enterprise Client version 11.x to Automation 360.

Request

```
POST https://{{ControlRoomURL}}/v3/migration/start
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request body:

```
{
  "name": "Migration.migrator.22.07.06.17.59.47",
  "description": "Migrating Legacy Bot",
  "overwriteBots": true,
  "botIds": [
    13
  ],
  "userIds": [
    5
  ],
  "legacyExcelCellRow": true,
  "includeChildFolders": false,
  "folderIds": [

  ],
  "emailEwsSettings": true,
  "emailEwsExchangeVersion": "Exchange2010",
  "emailEwsAuthenticationType": "Basic",
  "convertToEdgeWithIeMode": true,
  "botAnalytics": true,
  "webServicesProxySettings": true,
```

```
"excludeBotDependencies":true
}
```

Note: Complete the prerequisite tasks before migrating Enterprise 11 bots to Automation 360.
[Prerequisite tasks for migrating bots](#)

Request parameters

Parameter	Type	Required	Description
name	String	Yes	The name for the migration entity.
description	String	No	A short description for the migration.
overwriteBots	Boolean	No	Set this parameter to <code>true</code> if you want to migrate already migrated bots again. The default value is <code>false</code> .
botIds	integer	Yes	Bot IDs to run the migration.
userIds	Integer	Yes	Bot Runner users' IDs where this migration will be run on.
legacyExcelCellRow	Boolean	No	Set this parameter to <code>true</code> if you want to use the "Excel Cell Row" legacy behavior. If you are migrating from 11.3.0 or earlier, select this option. The default value is <code>false</code> .
includeChildFolders	Boolean	No	Set this parameter to <code>true</code> if you want to include the child folders during the migration. The default value is <code>false</code> .

Parameter	Type	Required	Description
folderIds	Integer	No	Folder IDs for all the bots contained in the subfolder with the <i>folderIds</i> .
emailEwsSettings	Boolean	No	Set this parameter to <code>true</code> if you have used EWS in Enterprise 11. The default value is <code>false</code> .
emailEwsExchangeVersion	String	No	This is used to specify the version of the EWS server configured in Enterprise 11. When you set <i>emailEwsSettings</i> to <code>true</code> , you can set a value to the email exchange server (<i>emailEwsExchangeVersion</i>). The possible values are as follows: <i>Exchange2010</i> , <i>Exchange2010_SP1</i> , <i>Exchange2010_SP2</i> , <i>Exchange2007_SP1</i> , and <i>Exchange2013</i> .
emailEwsAuthenticationType	String	No	This is used to specify the authentication type set for the EWS server configured in Enterprise 11. The possible values are as follows: <i>Basic</i> and <i>OAuth2</i> .
convertToEdgeWithIeMode	Boolean	No	Set this parameter to <code>true</code> if you want to migrate bots that use Internet Explorer to use Microsoft Edge with IE mode. The default value is <code>false</code> .
botAnalytics	Boolean	No	Set this parameter to <code>true</code> if you want to tag bots and

Parameter	Type	Required	Description
			variables for analytics through Bot Insight. If you are not using Bot Insight in legacy bots, set it to <code>false</code> . The default value is <code>false</code> .
webServicesProxySettings	Boolean	No	Set this parameter to <code>true</code> if you want to migrate along with the web services proxy host and port. If this value is true, precreated global values will be available to the bot for connecting with the web services proxy. The default value is <code>false</code> .
excludeBotDependencies	Boolean	No	Set this to <code>true</code> if you want all the bot dependencies to be excluded during migration. The default value is <code>false</code> .

Note: View the migration status using [List migration results API](#).

Response

200 OK

The successful response includes a 200 success response and an empty body.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

See also

- [Migrate all bots in a sub-folder API](#)
- [Migration action mapping results API](#)

Migrate all bots in a sub-folder API

Migrate all the bots separately as well as from a given folder and all if its sub-folders in your Control Room repository.

Prerequisites

- Find the folder **ID** you want to migrate. The [List files and folders by workspace API](#) searches for files and folders in the private or public Control Room repositories. Filter the results to identify the folder ids to be used in the request body.
- For one or more users with a RUNTIME device license. Use **userIds** for registered users in the Control Room as unattended bot runners with a RUNTIME device license and registered device. [Search for users API](#)
- You must have an admin role to migrate bots and folders.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST method and endpoint URL: **<your_control_room_url>/v3/migration/start**

Request parameters:

Parameter	Required	Type	Description
folderIds	Yes	Integer	Folder IDs for all the bots contained in the sub-folder with the folderIds.
userIds	Yes	Integer	User IDs for an unattended Bot Runner user.
includeChildFolders	No	Boolean	Set this parameter to true if you want to include child folders for the migration. The default value is false.

This following request starts a migration for all the bots contained in the sub-folder with the folderIds: 7 and userIds: 18.

Request body:

```
{
  "name": "Follow a convention that is meaningful and easy to search.",
  "description": "Add a meaningful description.",
  "overwriteBots": true,
  "userIds": [
    18
  ],
  "folderIds": [
    7
  ],
  "includeChildFolders": true
}
```

3. Send the request.

The successful response includes a 200 success code and an empty body.

```
{ }
```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Next steps

You can view the status of the migration using the [List migration results API](#) API.

List migration results API

List the overall migration results for each migration you run. Filter by selected fields to get the specific results.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

You must have an admin role or a custom role with **View Migration** permission.

Supported filterable fields: Use the following filters in the request body to narrow the results.

- name: The migration name.
- status: The migration status.
 - Success
 - Skipped
 - Failed
- migrationType: The migration type: BOT, ROLE, or AUDIT_DATA.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST method and endpoint URL: `<your_control_room_url>/v3/migration/list`.

Request body:

This request searches for migrations that contain the word **HRBotMigration** in the name field that was started between the specified dates.

```
{
  "sort": [
    {
      "field": "startTime",
      "direction": "desc"
    }
  ],
  "filter": {
    "operator": "substring",
    "value": "HRBotMigration",
    "field": "name"
  },
  "page": {
    "offset": 0,
    "total": 100,
    "totalFilter": 100,
    "length": 200
  }
}
```

3. Send the request.

Response body:

The example response returned the migration name, startTime and endTime, migration status, migrationType and other details.

```
{
  "page": {
    "offset": 0,
    "total": 17,
    "totalFilter": 1
  },
  "list": [
    {
      "id": 3,
      "name": " HRBotMigration ",
      "startTime": "2021-01-20T14:26:27.347Z",
      "endTime": "2021-01-20T14:27:36.617Z",
      "createdBy": 1,
      "duration": "69s",
      "numSuccess": 1,
      "numFailed": 0,
      "numSkipped": 0,
      "numTotal": 1,
      "status": "SUCCESSFUL",
      "updatedOn": "2021-01-20T14:26:47.850Z",
      "updatedBy": 1,
      "durationMillis": 69270,
      "migrationType": "BOT"
    }
  ]
}
```

- ⚠ **Note:** There are some response fields that are not used for Enterprise 11 migration:
- duration: A legacy field that is no longer used for migration.
 - migrationType: It is used for Enterprise 10 migrations only. It is not used for Enterprise 11 migration.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Next steps

To view details about a specific migration, enter a specific migration id in the [Bot migration results by id API](#).

Bot migration results by id API

List bot migration results by a unique numeric identifier, migrationId and filter the results by selected fields.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- You must have an admin role or a custom role with **View Migration** permission.
- Use the numeric identifier, migrationId for the migration you want to view.

Supported filterable fields:

Use the following filters in the request body to narrow the search results:

- sourceName: Source bot name.
- sourceType: Source bot type.
- status: Migration status.
 - Success
 - Skipped
 - Failed
- reason: A reason why this bot migration is failed.
- targetName: Migrated bot target name.
- targetType: Migrated bot target type.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST method and endpoint URL: `<your_control_room_url>/v3/migration/<migrationId>/results/list`
Enter the migration ID you want to view.
3. Create a request to find the search results. This filter searches for a string in the sourceName of the migrated bot.

Request body:

```
{
  "sort": [
    {
      "field": "sourceName",
      "direction": "asc"
    }
  ],
  "filter": {
    "operator": "substring",
    "value": "mbot-dep",
    "field": "sourceName"
  },
  "page": {
    "offset": 0,
    "total": 100,
    "totalFilter": 100,
    "length": 200
  }
}
```

4. Send the request.

Response body:

```
{
  "page": {
    "offset": 0,
    "total": 3,
    "totalFilter": 1
  },
  "list": [
    {
      "sourceId": 24,
      "sourceName": "mbot-dep01.mbot",
      "sourcePath": "Automation Anywhere\\Bots\\My MetaBots\\mbot-dep01.mbot",
      "sourceType": "application/vnd.aa.mbot",

```



```

        "targetId": 941,
        "status": "SUCCESS",
        "reason": "",
        "selectedByUser": true,
        "userId": 9,
        "id": 469,
        "targetName": "logic-launchweb01",
        "targetPath": "Automation Anywhere\\Bots\\My MetaBots\\mbot-d
ep01\\logic-launchweb01",
        "targetType": "application/vnd.aa.taskbot"
    }
]
}

```

The response returned 1 out of 3 responses for bot migration results.

add link to the next API Migration Action mapping results.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Next steps

To view the list action mapping results, see [Migration action mapping results API](#).

Migration action mapping results API

List action mapping results for bots by unique numeric identifiers for the migration <migration ID> and the journal <journal ID>, and then filter the results by selected fields.

Request

```
POST https://{ControlRoomURL}/v3/migration/<migration ID>/journal/<journal ID>/actionmappings/list
```

```
Header: X-Authorization: <<authentication token>> or Authorization: Bearer <<bearer token>>
```

All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.

Request body

```
{
  "sort": [
    {
      "field": "targetLineNumber",
      "direction": "asc"
    }
  ],
  "filter": {
    "field": "reviewRequired",
    "operator": "eq",
    "value": "true"
  },
  "page": {
    "offset": 0,
    "length": 100
  }
}
```

Make sure you have the following:

- You have an admin role or a custom role with **View migration** permission.
- The numeric identifier, <migration ID>, for the migration you want to view.
- The numeric value for the <journal ID> associated with the migration identifier. See [How to find a migration journalid](#)

Request parameters

Parameter	Type	Required	Description
sort	Array	No	<p>By default, search results are sorted in descending order with respect to their IDs. An alternative sorting is specified using the sort query parameter.</p> <p>Enter the field by which you want to sort along with the direction <code>asc</code> (ascending) or <code>desc</code> (descending).</p>

Parameter	Type	Required	Description
filter	Object	No	Filters the result.
page	Object	No	The page object allows you to get the desired pages.

For more information on filtering, pagination, and sorting, see [Filtering, pagination, and sorting](#).

Response

```
{
  "page": {
    "offset": 0,
    "total": 1,
    "totalFilter": 1
  },
  "list": [
    {
      "targetLineNumber": 1,
      "targetAction": "TerminalEmulator-connectV2",
      "isReviewRequired": true,
      "reason": "'Terminal Emulator - Connect' action has been migrated.\n\nThe connection type is configured as 'SSH2' since 'SSH1' is not supported in A 360.\n\nNo further action required.\n\nMessage Code: R114",
      "remarks": "",
      "id": 1309,
      "sourceLineNumber": 1,
      "sourceAction": "TE-Connect",
      "targetNodeId": "ba97973a-e231-44b3-b208-1056dc2520d8",
      "targetPackageName": "TerminalEmulator",
      "targetPackageVersion": "4.3.0-20211016-070439",
      "isActionRequired": false,
      "actionRequiredReason": "",
      "reviewRequiredReasonMarkdown": "",
      "actionRequiredReasonMarkdown": ""
    }
  ]
}
```

```

    }
  ]
}
```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Response parameters

Parameter	Type	Description
targetLineNumber	Integer	The target line number in the migrated bot.
targetAction	String	The target action in the migrated bot.
isReviewRequired	Boolean	Specifies whether the action migrated requires your review.
reason	String	Describes the reason for the review required.
remarks	String	Provides more information on this migrated action.
id	Integer	ID of the migration action mapping result.
sourceLineNumber	Integer	Source line number in source bot.
sourceAction	String	Source action in source bot
targetNodeId	String	Target node UID
targetPackageName	String	Target action package name
targetPackageVersion	String	Target action package version

Parameter	Type	Description
isActionRequired	Boolean	Specifies whether the action migrated requires your action
actionRequiredReason	String	Describes what action is required
reviewRequiredReasonMarkdown	String	Review required reason with support help link for the message code
actionRequiredReasonMarkdown	String	Action required reason with support help link for the message code

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

How to find a migration journalid

Migrations can have more than one journalid. You can find the journalid in the response of a results list for a specific migration id.

In this request, we searched for results for the migration with the id **16**.

```
<your_control_room_url>/v3/migration/16/results/list
```

The request returned the id in each object list as the journalid. In this example, there are two journal identifiers, 30 and 31, for the migration id 16.

```
{
  "page": {
    "offset": 0,
    "total": 2,
    "totalFilter": 2
  },
  "list": [
    {
```

```

        "sourceId": 12,
        "sourceName": "Dependency of IGN-23437.mbot",
        "sourcePath": "Automation Anywhere\\Bots\\My MetaBots\\Dependency
of IGN-23437.mbot",
        "sourceType": "application/vnd.aa.mbot",
        "targetId": 0,
        "status": "FAILED",
        "reason": "The logic IGN-23437 has some commands or actions which
are not yet supported for migration.",
        "selectedByUser": true,
        "userId": 9,
        "id": 30,
        "targetName": "",
        "targetPath": "",
        "targetType": ""
    },
    {
        "sourceId": 12,
        "sourceName": "Dependency of IGN-23437.mbot",
        "sourcePath": "Automation Anywhere\\Bots\\My MetaBots\\Dependency
of IGN-23437.mbot",
        "sourceType": "application/vnd.aa.mbot",
        "targetId": 0,
        "status": "FAILED",
        "reason": "Migration of MetaBot failed.",
        "selectedByUser": false,
        "userId": 9,
        "id": 31,
        "targetName": "",
        "targetPath": "",
        "targetType": ""
    }
]
}

```

You can use the migration id and journal id in an action mapping request.

```
<your_control_room_url>/v3/migration/16/journal/31/actionmappings/list
```

Enterprise 10 Migration APIs

Use migration APIs to migrate MetaBots and TaskBots that were created in Enterprise Client version Enterprise 10 to Automation 360. With these APIs, you can connect to the Enterprise 10 Control Room database, validate the master key and the repository path, and then start copying the Enterprise 10 data to Automation 360.

- Complete the prerequisites before migrating Enterprise 10 bots to Automation 360. See [Prerequisite tasks for migrating bots](#).
- You must have an admin role or the Manage Migration permission to validate connection parameters, connect to the Enterprise 10 database, and start 10.x migration process. If you want to retrieve a list of roles, users, credentials, and schedules, you need an admin role or the View Migration permission.
- Execute the following three APIs in order they are listed below. You can retrieve roles, users, credentials, bots, or schedules after you executed the mandatory APIs.

Connect to Enterprise 10 database

Use this API to connect to the Enterprise 10 Control Room database from which you can copy the data to Automation 360.

Prerequisites

Note: You can view the [Control Room APIs](#) in the Community Edition, but API functionality is limited. You need a licensed Automation 360 Edition to access the full functionality of the APIs.

- Review the prerequisites before migrating bots to Automation 360.

[Prerequisite tasks for migrating bots](#)

- You must have an admin role or a custom role with **Manage Migration** permission to validate connection parameters and connect to the Enterprise 10 database.

Procedure

1. All API calls must contain either an authentication token from [Authentication API](#) (generates JSON Web token) or a bearer token from OAuth services. You cannot use both together in an API.
2. Use the POST method and endpoint URL: `<your_control_room_url>/v2/migration/connection`

Request body:

```
{
    "host": "10.000.000.000",
    "port": 1433,
    "databaseName": "CRDB",
    "username": "Admin",
    "password": "<password>",
    "integratedSecurity": false,
    "encrypt": false
}
```

Request parameters:

Parameter	Required	Type	Description
host	Yes	String	SQL server host name or IP address
port	Yes	Integer	SQL server port number
databaseName	Yes	String	Source Control Room database name
userName	Yes	String	A user name to connect to the database
password	Yes	String	Password to connect to the database
integratedSecurity	Yes	Boolean	Set this to <code>true</code> if you want use Windows authentication. The default value is <code>false</code> .
encrypt	Yes	Boolean	Set this to <code>true</code> if you want to use a secure connection to the source database. The default value is <code>false</code> .

3. Send the request.

Response body:

It returns a success response code and an empty string.

```
{ }
```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Next steps

[Validate master key and repository path](#)

Validate master key and repository path

Use this API to validate the Credential Vault master key and the repository path.

Prerequisites

- You must have an admin role or a custom role with **Manage Migration** permission to validate the master key and the repository path.
- Complete this step [Connect to Enterprise 10 database](#) before executing this API.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the POST method and endpoint URL: `<your_control_room_url>/v2/migration/masterkey`

Request body:

```
{
  "privateKey": "<CV master key value>",
  "repoPath": "C:\\Migration\\10X\\A2019.14\\Automation Anywhere Server
Files"
}
```

If you are not able to get a response, add an additional backslash in the repoPath or use a single forward slash.

Request parameters:

Parameter	Required	Type	Description
privateKey	Yes	String	The master key to connect to Enterprise 10 Credential Vault. This is available for configuration during the initial Control Room setup.
repoPath	Yes	String	The shared repository path where the Control Room Enterprise 10 repository is located.

3. Send the request.

Response body:

It returns a success response code and an empty string.

```
{ }
```

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Next steps

[Initiate the 10.x data migration process](#)

Initiate Enterprise 10 data migration process

Use this API to start the Enterprise 10 data migration.

Prerequisites

- You must have an admin role or a custom role with **Manage Migration** permission to start the Enterprise 10 data migration.
- Complete this step [Validate master key and repository path](#) before executing this API.

Procedure

- Add the authentication token or bearer token to the request header. You cannot use both in the same API.
- Use the POST method and endpoint URL: `<your_control_room_url>/v2/migration/start`

Request body:

```
{
  "name": "Migration-10x-001",
  "description": "10x bot migration to A2019 001"
}
```

Request parameters:

Parameter	Required	Type	Description
name	Yes	String	The name for the migration entity.
description	No	String	A short description for the migration.

3. Send the request.

Response body:

```
{
  "id": "1",
  "name": "Migration-10x-001",
  "createdBy": "10",
  "migrationType": "ROLE",
  "entities": []
}
```

Response parameters:

Parameter	Description
id	The migration ID This is used for internal purpose only.
name	The migration name.
createdBy	ID of the user who has initiated this migration.

Parameter	Description
migrationType	Indicates the data migration type: Role or Bot This is used for Enterprise 10 migration only.
entities	List of migration entities.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Next steps

[Retrieve a list of migration roles](#)

Retrieve migrated roles

Use this API to retrieve Enterprise 10 roles that are copied to the Automation 360 database.

Prerequisites

You must have an admin role or a custom role with **View Migration** permission to retrieve migrated roles.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the ID from the API response body as the migrationId.
See [Initiate Enterprise 10 data migration process](#).
3. Use the POST method and endpoint URL: `<your_control_room_url>/v2/migration/{migrationId}/roles/list`

Request body:

```
{
  "sort": [],
  "filter": {},
  "page": {
    "offset": 0,
    "total": 100,
    "totalFilter": 100,
    "length": 200
  }
}
```

```
}  
}
```

Request parameters:

Parameter	Required	Type	Description
sort	No	String	Sort directions
filter	No	String	Filter rules
page	Yes	Integer	Pagination rules
offset	Yes	Integer	Page starting index
total	No	Integer	Total number of items
totalFilter	No	Integer	Total number of items matching the filter
length	Yes	Integer	Number of items to be returned

4. Send the request.

Response body:

```
{  
  "page": {  
    "offset": 0,  
    "total": 9,  
    "totalFilter": 9  
  },  
  "list": [  
    {  
      "id": 1,  
      "type": "ROLE",  
      "sourceId": "1",  
      "targetId": 1,  
    }  
  ]  
}
```

```
    "name": "Admin",
    "status": "SUCCESS",
    "reason": "",
    "targetPath": ""
  },
  {
    "id": 2,
    "type": "ROLE",
    "sourceId": "2",
    "targetId": 2,
    "name": "Basic",
    "status": "SUCCESS",
    "reason": "",
    "targetPath": ""
  },
  .....
  {
    "id": 9,
    "type": "ROLE",
    "sourceId": "9",
    "targetId": 24,
    "name": "Admin_Role01",
    "status": "SUCCESS",
    "reason": "",
    "targetPath": ""
  }
]
```

Response parameters:

Parameter	Description
list	List of roles.
id	The role ID.

Parameter	Description
	This ID is used for UI purposes only.
type	The migration type.
sourceId	The source (database) role ID.
targetId	The migrated role ID.
name	A role name.
status	The migration status Valid values: Success, Skipped, or Failed.
reason	The reason why the migration failed.
targetPath	The migrated bot file location. It is only applicable for the bot migration.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Next steps

[Retrieve a list of migrated users](#)

Retrieve migrated users

Use this API to retrieve Enterprise 10 database users that are copied to the Automation 360 database.

Prerequisites

You must have an admin role or a custom role with **View Migration** permission to retrieve migrated users.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the ID from the API response as the migrationId.
See [Initiate Enterprise 10 data migration process](#).
3. Use the POST method and endpoint URL: `<your_control_room_url>/v2/migration/{migrationId}/users/list`

Request body:

```
{
  "sort": [],
  "filter": {},
  "page": {
    "offset": 0,
    "total": 100,
    "totalFilter": 100,
    "length": 200
  }
}
```

Request parameters:

Parameter	Required	Type	Description
sort	No	String	Sort directions
filter	No	String	Filter rules
page	Yes	Integer	Pagination rules
offset	Yes	Integer	Page starting index
total	No	Integer	Total number of items
totalFilter	No	Integer	Total number of items matching filter

Parameter	Required	Type	Description
length	Yes	Integer	Number of items to be returned

4. Send the request.

Response body:

```
{
  "page": {
    "offset": 0,
    "total": 5,
    "totalFilter": 5
  },
  "list": [
    {
      "id": 10,
      "type": "USER",
      "sourceId": "1",
      "targetId": 11,
      "name": "admin_1",
      "status": "SUCCESS",
      "reason": "The user admin has been renamed to admin_1 as the user with same name already exists",
      "targetPath": ""
    },
    .....
    {
      "id": 14,
      "type": "USER",
      "sourceId": "5",
      "targetId": 15,
      "name": "admin10503",
      "status": "SUCCESS",
      "reason": "",
      "targetPath": ""
    }
  ]
}
```

```
]
}
```

Response parameters:

Parameter	Description
list	List of users
id	The user ID It is used for UI purpose only.
type	The migration type
sourceId	The source (database) user ID
targetId	The migrated user ID
name	A user name
status	The migration status Valid values: Success, Skipped, or Failed.
reason	The reason why the migration failed.
targetPath	The migrated bot file location It is only applicable for bot migration.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Next steps

[Retrieve a list of migrated credentials](#)

Retrieve migrated credentials

Use this API to retrieve Enterprise 10 database credentials that are copied to the Automation 360 database.

Prerequisites

You must have an admin role or a custom role with **View Migration** permission to retrieve migrated credentials.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API. Use the HTTP request syntax (URL):
2. Use the ID from the API response body as the migrationId.
See [Initiate Enterprise 10 data migration process](#).
3. Use the POST method and endpoint URL: `<your_control_room_url>/v2/migration/{migrationId}/credentials/list`

Request body:

```
{
  "sort": [],
  "filter": {},
  "page": {
    "offset": 0,
    "total": 100,
    "totalFilter": 100,
    "length": 200
  }
}
```

Request parameters:

Parameter	Required	Type	Description
sort	No	String	Sort directions
filter	No	String	Filter rules

Parameter	Required	Type	Description
page	No	Integer	Pagination rules
offset	Yes	Integer	Page starting index
total	No	Integer	Total number of items
totalFilter	No	Integer	Total number of items matching the filter
length	Yes	Integer	Number of items to be returned

4. Send the request.

Response body:

```
{
  "page": {
    "offset": 0,
    "total": 2,
    "totalFilter": 2
  },
  "list": [
    {
      "id": 15,
      "type": "CREDENTIAL",
      "sourceId": "4",
      "targetId": 2,
      "name": "admin10501 - Email Settings",
      "status": "SUCCESS",
      "reason": "",
      "targetPath": ""
    },
    {
      "id": 16,
      "type": "CREDENTIAL",
```

```
{
  "sourceId": "5",
  "targetId": 2,
  "name": "admin10502 - Email Settings",
  "status": "SUCCESS",
  "reason": "",
  "targetPath": ""
}
```

Response parameters:

Parameter	Description
list	List of credentials
id	The credential ID
type	The migration type
sourceId	The source (database) credential ID
targetId	The migrated credential ID
name	A credential name
status	The migration status Valid values: Success, Skipped, or Failed.
reason	The reason why the migration failed
targetPath	The migrated bot file location It is only applicable for the bot migration.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Next steps

[Retrieve a list of migrated bots](#)

Retrieve migrated bots

Use this API to retrieve a list of Enterprise 10 migrated bots that are copied to the Automation 360 database.

Prerequisites

You must have an admin role or a custom role with **View Migration** permission to retrieve migrated bots.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API. Use the HTTP request syntax (URL):
2. Use the ID from the API response body as the migrationId.
See [Initiate Enterprise 10 data migration process](#).
3. Use the POST method and endpoint URL: `<your_control_room_url>/v2/migration/{migrationId}/bots/list`

Request body:

```
{
  "sort": [],
  "filter": {},
  "page": {
    "offset": 0,
    "total": 100,
    "totalFilter": 100,
    "length": 200
  }
}
```

Request parameters:

Parameter	Required	Type	Description
sort	No	String	Sort directions

Parameter	Required	Type	Description
filter	No	String	Filter rules
page	No	Integer	Pagination rules
offset	Yes	Integer	Page starting index
total	No	Integer	Total number of items
totalFilter	No	Integer	Total number of items matching the filter
length	Yes	Integer	Number of items to be returned

4. Send the request.Send the request.

Response body:

```
{
  "page": {
    "offset": 0,
    "total": 1004,
    "totalFilter": 1004
  },
  "list": [
    {
      "id": 17,
      "type": "BOT",
      "sourceId": "186",
      "targetId": 15,
      "name": "DelayLoop - Copy (129).atmx",
      "status": "SUCCESS",
      "reason": "",
      "targetPath": "Automation Anywhere\\Bots\\My Tasks\\Migration Extra
    },
  ]
}
```

```
{
  "id": 18,
  "type": "BOT",
  "sourceId": "438",
  "targetId": 17,
  "name": "DelayLoop - Copy (115).atmx",
  "status": "SUCCESS",
  "reason": "",
  "targetPath": "Automation Anywhere\\Bots\\My Tasks\\Migration Extra
\\RM-APIAutomation"
},
.....
{
  "id": 216,
  "type": "BOT",
  "sourceId": "703",
  "targetId": 221,
  "name": "DelayLoop - Copy (132).atmx",
  "status": "SUCCESS",
  "reason": "",
  "targetPath": "Automation Anywhere\\Bots\\My Tasks"
}
]
}
```

Response parameters:

Parameter	Description
list	List of bots
id	The bot ID
type	The migration type

Parameter	Description
sourceId	The source (database) bot ID
targetId	The migrated bot ID
name	The bot name
status	The migration status Valid values: Success, Skipped, or Failed.
reason	The reason why the migration failed
targetPath	The migrated bot file location

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Next steps

[Retrieve a list of migrated schedules](#)

Retrieve migrated schedules

Use this API to retrieve a list of Enterprise 10 migrated schedules that are copied to the Automation 360 database.

Prerequisites

You must have an admin role or a custom role with **View Migration** permission to retrieve migrated schedules.

Procedure

1. Add the authentication token or bearer token to the request header. You cannot use both in the same API.
2. Use the ID from the API response body as the migrationId .
See [Initiate Enterprise 10 data migration process](#).

3. Use the POST method and endpoint URL: `<your_control_room_url>/v2/migration/{migrationId}/schedules/list`

Request body:

```
{
  "sort": [],
  "filter": {},
  "page": {
    "offset": 0,
    "total": 100,
    "totalFilter": 100,
    "length": 200
  }
}
```

Request parameters:

Parameter	Required	Type	Description
sort	No	String	Sort directions
filter	No	String	Filter rules
page	No	Integer	Pagination rules
offset	Yes	Integer	Page starting index
total	No	Integer	Total number of items
totalFilter	No	Integer	Total number of items matching the filter
length	Yes	Integer	Number of items to be returned

4. Send the request.

Response body:

```
{
  "page": {
    "offset": 0,
    "total": 7,
    "totalFilter": 7
  },
  "list": [
    {
      "id": 1173,
      "type": "SCHEDULE",
      "sourceId": "3",
      "targetId": 2,
      "name": "Monthly",
      "status": "SUCCESS",
      "reason": "",
      "targetPath": ""
    },
    {
      "id": 1174,
      "type": "SCHEDULE",
      "sourceId": "5",
      "targetId": 3,
      "name": "alternatemonths",
      "status": "SUCCESS",
      "reason": "",
      "targetPath": ""
    },
    .....
    {
      "id": 1179,
      "type": "SCHEDULE",
      "sourceId": "15",
      "targetId": 8,
      "name": "none",
      "status": "SUCCESS",
      "reason": "",
```

```
"targetPath": ""  
}  
]  
}
```

Response parameters:

Parameter	Description
list	List of schedules
id	The schedule ID
type	The migration type
sourceId	The source (database) schedule ID
targetId	The migrated schedule ID
name	The schedule name
status	The migration status Valid values: Success, Skipped, or Failed.
reason	The reason why the migration failed
targetPath	The migrated bot file location It is only applicable for the bot migration.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Filtering, pagination, and sorting

The Control Room API supports filtering, pagination, and sorting for endpoints that return arrays of resources.

The filtering mechanism filters the required resources, the sorting mechanism places the resources in order; and the pagination mechanism then returns a specific range of those ordered resources. This topic provides you the details to filter and sort the results of an API requests and also guides to handle the pagination of large result sets returned from an API request.

Note:

- Sorting and filtering are supported for substrings. For example, if you want to search for bots or files that have `fin` in their names, enter `fin` as the search criterion. All the bots and files that contain `fin` in the names will be displayed, for example, Finance, Finder, DeltaFinance, and Dolfin.
- Wildcards are not supported for searching and filtering bots or files.

Filtering

Filtering allows you to apply Boolean condition against a collection of returned resources in order to subset the collection to only those resources for which the condition is `true`. The most basic operation in an Control Room API filters is to compare a field to a given value. It is possible to use *equality comparison*, *range comparison*, or *logical*. Use the following operators to compare a field to a constant value.

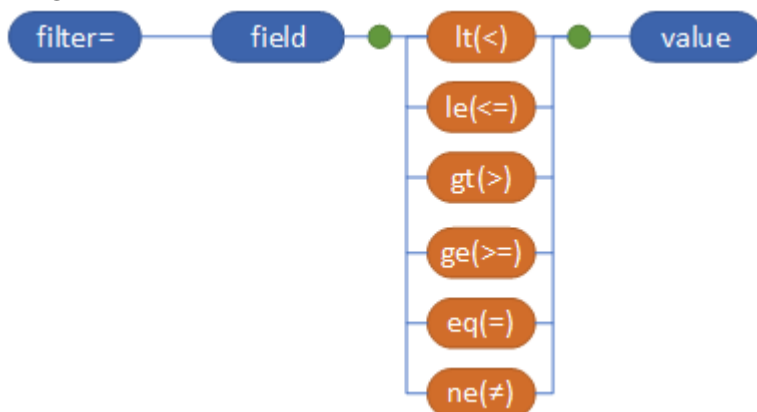
Operation	Description	Example
Equality comparison		
eq	Equals	UserEmailAddress, eq first.last@aa.com
ne	Not Equals	UserEmailAddress, ne first.last@aa.com
Range comparison		
lt	Less than	Quantity lt 1500
le	Less Than or Equal	Quantity le 1500

Operation	Description	Example
ge	Greater Than or Equal	CreatedDateUtc ge 2021-03-15
gt	Greater Than	CreatedDateUtc gt 2021-03-15
Logical		
and	And	Field1 eq 'abc' and Field2 eq 'def'
or	Or	Field1 eq 'abc' or Field2 eq 'def'

The **filter** query parameter allows you to apply basic, multiple, and convention oriented filters to a request. The filters in the Control Room APIs are applied with single parameter or with multiple parameters.

Single parameter filter

Single parameter filter allows the API request to select the responses by matching one or more members of the response to the value passed in the query. The single parameter filter is represented in the following image:



The JSON equivalent of the above image (single parameter filter) looks like:

```

{
  "filter": {
    "operator": "<NONE, lt, le, eq, ne, ge, gt, substring, and, or, not>",
    "field": "string",
  }
}
  
```

```

    "value": "string"
  }
}

```

For example, to list all the device pools that has a substring `finance`, use the following single parameter filter:

```
POST http://{{ControlRoomURL}}/v2/devices/pools/list
```

```

{
  "filter":{
    "operator":"substring",
    "field":"name",
    "value":"finance"
  }
}

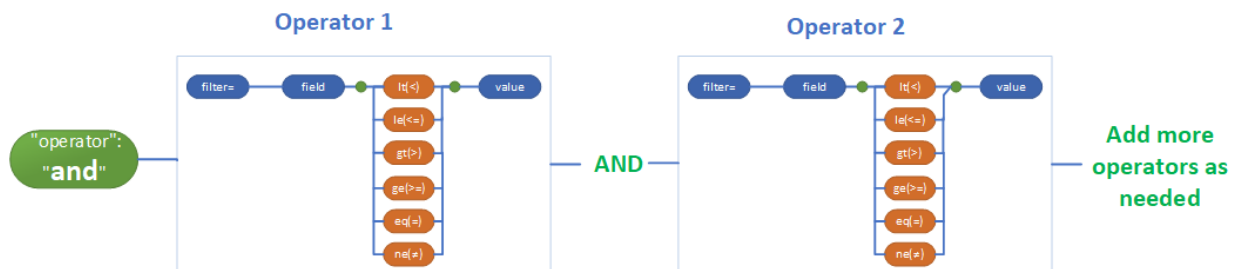
```

For more detailed sample on a single parameter filter, see [List device pools API](#).

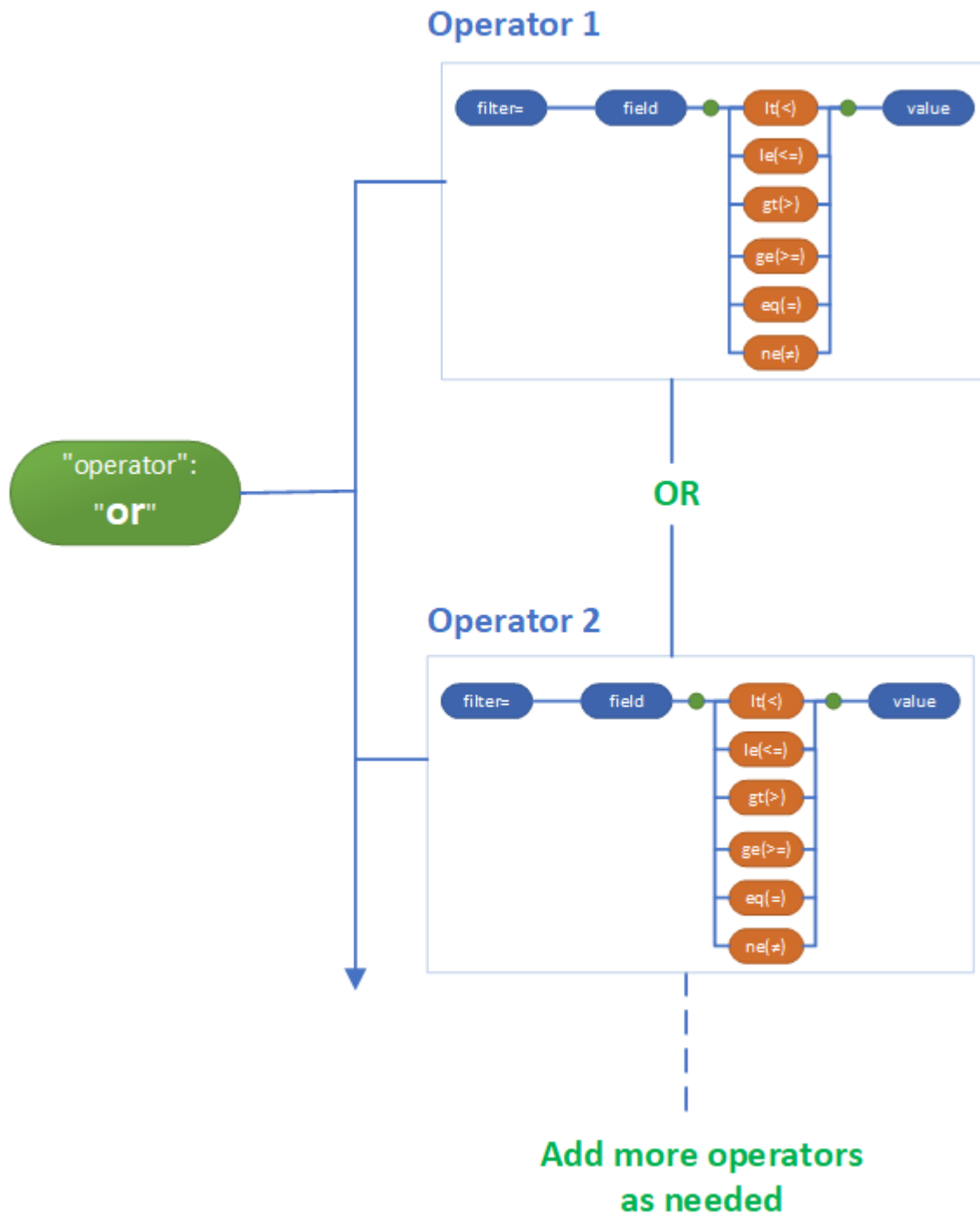
Multiple parameter filter

Multiple parameter filter allows you to filter the results based on combining multiple conditions wrapped in logical operands `and` and `or`.

- `and`: A binary operator that evaluates to `true` if all the conditions in the operands evaluate to `true`.



- `or`: A binary operator that evaluates to `true` if atleast one of the conditions in the operands evaluate to `true`.



The JSON equivalent of the above image (multiple parameter filter) looks like:


```
{
  "filter": {
    "operator": "<and, or>",
    "operands": [
      {
        "operator": "<NONE, lt, le, eq, ne, ge, gt, substring, and, or, not>",
        "field": "string",
        "value": "string"
      },
      {
        "operator": "<NONE, lt, le, eq, ne, ge, gt, substring, and, or, not>",
        "field": "string",
        "value": "string"
      }
    ]
  }
}
```

For example, to list all the roles that has a substring `Device` , `createdOn` is after `2022-04-01` , and `createdOn` is before `2022-05-31` , use the multiple parameter filtering with logical `add` operator as follows:

```
POST http://{{ControlRoomURL}}/v1/usermanagement/roles/list
```

```
{
  "filter":{
    "operator":"and",
    "operands":[
      {
        "operator":"substring",
        "field":"name",
        "value":"Device"
      },
      {
        "operator":"gt",
```

```

        "field": "createdOn",
        "value": "2022-04-01T00:00:00.989Z"
      },
      {
        "operator": "lt",
        "field": "createdOn",
        "value": "2022-05-31T23:00:00.123Z"
      }
    ]
  }
}

```

For more detailed sample on a single parameter filter, see [List roles](#).

Pagination

Pagination allows you to:

- Retrieve a limited collection of results.
- Offset a collection of results.

All Control Room APIs that returns a collection of records are paginated. API methods that support pagination takes two (optional) parameters:

Operation	Description
offset	The offset parameter controls the starting point within the collection of response results. Default value is 0.
length	The length parameter is the maximum number of records to retrieve starting from the offset. Default value is 200.

The JSON snippet for pagination looks like:

```

"page": {
  "offset": 5,
  "length": 10
}

```

For more detailed sample on a single parameter filter, see [List roles](#).

Sorting

Sorting allows you to order the results by any field, in ascending or descending order. For example, if you are returning the roles, you can sort the roles by last modified date.

```
"sort": [
  {
    "field": "string",
    "direction": "<asc, desc>"
  }
]
```

Direction

Type: Enum [desc, asc]

- asc = ascending (smallest to largest, 0 to 9, A to Z)
- desc = descending (largest to smallest, 9 to 0, Z to A)

For more detailed sample on a single parameter filter, see [List roles](#).

API response codes

Review the HTTP status codes of responses for Automation 360 APIs.

Status code	HTTP name	Description
200	OK	Success
201	Created	Success, for POST, PATCH or PUT requests.
202	Accepted	Indicates that the request has been accepted for processing, but the processing has not been completed.
204	No content	Success, for DELETE requests and some PATCH requests.

Status code	HTTP name	Description
400	Bad request	Request URL or request parameters are incorrect.
401	Authentication required	Provide authentication details.
403	Unauthorized access	The operation is not authorized.
404	Not found	Control Room server did not find the requested URL.
409	Conflict	The request could not be completed because the code conflicts with the current state of the resource, for example, a duplicate entry.
500	Internal server error	Indicates that the server encountered a problem. Clear the cookies and cache, and then reload the page.

Parent topic: [Control Room APIs](#)

Comparing Automation 360 and Enterprise 11 APIs

Compare Automation 360 and Enterprise 11 APIs to understand the contract changes when you migrate from Enterprise 11 to Automation 360.

API details	Enterprise 11	Automation 360
Authentication API: Use this API to obtain the authentication token. The token is then used for all subsequent API calls.		
Method	POST	POST
Endpoint	<code>v1/authentication</code>	<code>v1/authentication</code>

API details	Enterprise 11	Automation 360
Request body changes	Not applicable	No change
Response body changes	Not applicable	<ul style="list-style-type: none"> Multi-factor authentication is not supported Automation 360 response returns the <code>tenantUUID</code>
Auto login credentials API: Use this API to set the Windows credentials for a Bot Runner. These credentials are used for automatically login (auto-login).		
Method	POST	PUT
Endpoint	<code>v1/credentialvault/external/credentials/loginsetting</code>	<code>v2/credentialvault/loginsetting</code>
Request body changes	Not applicable	Endpoint version change
Response body changes	Not applicable	<p>Message displayed after successfully updating the credentials: Credentials updated for <username>.</p> <p>Enterprise 11 displays this message: Credentials were successfully updated.</p>
Automation management API: Use this API to create automations (schedule bots), and edit and delete automations.		

API details	Enterprise 11	Automation 360
Method	POST, PUT, and DELETE	POST, PUT, GET, and DELETE
Endpoint	<code>v1/schedule</code>	<code>v1/schedule</code>
Request body changes	Not applicable	Minor change in request body to pass bot input variables
Response body changes	Not applicable	Detailed response with bot input variable details
User Management API: Use this API to create, edit, and delete users and roles.		
Method	POST, PUT, GET, and DELETE	POST, PUT, GET, and DELETE
Endpoint	<code>v1/usermanagement</code>	<code>v1/usermanagement</code>
Request body changes	Not applicable	No change
Response body changes	Not applicable	No change
Migration API: Use this API to migrate bots from Enterprise 11 or Enterprise 10 to Automation 360		
Method	GET and POST	GET and POST
Endpoint	<ul style="list-style-type: none"> <code>v1/migration</code> <code>v2/migration</code> 	<code>v2/migration</code>
Response body changes	Not applicable	<ul style="list-style-type: none"> Endpoint version change

API details	Enterprise 11	Automation 360
		<ul style="list-style-type: none"> APIs used to connect to the Enterprise 10 database to migrate users, roles, schedules, and bots
Request body changes	Not applicable	Because the endpoints and request body are different, the response body is also different.
Manual Dependency API: Use this API to add or remove a dependency (data files and such) to or from a bot.		
Method	POST and DELETE	Not available
Endpoint	<code>v1/files/manualdependencies/</code>	Not available
Request body changes	Not applicable	Not available
Response body changes	Not applicable	Not available
Repository Management API: Use this API to get bots and files from the Control Room repository.		
Method	POST, DELETE, and GET	POST, DELETE, and GET
Endpoint	<code>v1/repository</code>	<code>v1/repository</code>
Request body changes	Not applicable	Additional APIs added to get folder permissions

API details	Enterprise 11	Automation 360
Response body changes	Not applicable	No change
File Dependency API: Use this API to get the file dependency metadata to run and schedule bots.		
Method	GET	Not available
Endpoint	<code>v1/files/manualdependencies/</code>	Not available
Request body changes	Not applicable	Not available
Response body changes	Not applicable	Not available
Bot Lifecycle Management (BLM) API: Use this API to move (export or import) bots and dependent files across different Control Room environments.		
Method	POST	POST
Endpoint	<code>v1/blm</code>	<code>v2/blm</code>
Request body changes	Not applicable	<ul style="list-style-type: none"> Endpoint version change You cannot choose dependencies when exporting and importing
Response body changes	Not applicable	No change

API details	Enterprise 11	Automation 360
Audit API: Use this API to get audit information about the product.		
Method	POST	POST
Endpoint	<code>v1/audit</code>	<code>v1/audit</code>
Request body changes	Not applicable	No change
Response body changes	Not applicable	No change
Two-factor Authentication (2FA) API: Use this API to generate a 2FA token		
Method	GET and POST	Not available
Endpoint	<code>v1/mfa</code>	Not available
Request body changes	Not applicable	Not available
Response body changes	Not applicable	Not available
Credential Vault API: Use this API to create, edit, and delete credentials and lockers.		
Method	POST, PUT, GET, and DELETE	POST, PUT, GET, and DELETE
Endpoint	<code>v2/credentialvault</code>	<code>v2/credentialvault</code>
Request body changes	Not applicable	No change

API details	Enterprise 11	Automation 360
Response body changes	Not applicable	No change
Bot Execution Orchestration API: Use this API to get the repository, automations, and devices list.		
Method	POST	POST, PUT, GET, and DELETE
Endpoint	<code>v2/automations/deploy</code>	<code>v3/automations/deploy</code>
Request body changes	Not applicable	Bot deployment includes run-as users
Response body changes	Not applicable	Returns <code>deploymentID</code> . Enterprise 11 API response returns <code>automationID</code> .
License API: Use this API to get product license information.		
Method	GET	GET
Endpoint	<code>v2/license</code>	<code>v2/license</code>
Request body changes	Not applicable	No change
Response body changes	Not applicable	Additional APIs added to update with license server and additional details
Workload Management (WLM) API: Use this API to create, edit, and delete workload queues, templates, and work items.		

API details	Enterprise 11	Automation 360
Method	POST, PUT, GET, and DELETE	POST, PUT, GET, and DELETE
Endpoint	<code>v2/wlm</code>	<code>v3/wlm</code>
Request body changes	Not applicable	No change
Response body changes	Not applicable	More APIs to manage WLM entities
Bot Insight API: Use this API to get Bot Insight data for an automation.		
Method	POST, GET, and DELETE	POST, GET, and DELETE
Endpoint	<code>v2/botinsight/data/api</code>	<code>v2/botinsight/data/api</code>
Request body changes	Not applicable	No change
Response body changes	Not applicable	No change
Bot Insight JSON API: Use to get business insights for an automation.		
Method	GET	Not applicable
Endpoint	<code>v2/botinsight/data/api</code>	Not applicable
Request body changes	Not applicable	Not applicable
Response body changes	Not applicable	Merged with Bot Insight API

Parent topic: [Control Room APIs](#)

Bot Agent API: Auto registration

Automatically register and connect your device to a Control Room by using the Auto registration API.

This API uses the generic registration token from the auto-registration.properties file to register the device in the specified Control Room URL. You cannot autoregister the device if the Control Room URL is not available in the auto-registration.properties file. The auto-registration.properties file must be available on your local system, and you must not delete the file after the registration is complete.

Note: To register the device from the command line or API, you must set the AA_DELAY_REGISTRATION_UNTIL_LOGIN MSI parameter.

Request

```
POST http://127.0.0.1:22113/v1/registration/auto
```

Request body:

```
{
  "url": "https://{controlroom url}",
  "userName": "dpcreator"
}
```

Request parameters

Parameter	Type	Required	Description
url	String	Yes	Specify the Control Room URL to autoregister the device.
userName	String	No	Specify the Control Room user to associate the device as the default device.

Response

```
{
  "result": "REGISTERED",
  "deviceId": "11",
  "crUrl": "control_room_url",
  "userName": "dpcreator"
  "installationType": "SYSTEM_WIDE",
  "crSwitchAllowed": "false",
}
```

Response parameters

Parameter	Type	Description
result	String	Status of the device registration.
deviceId	Number	Unique ID of the device the Bot Agent is installed on.
crUrl	String	The Control Room URL where the device is autoregistered.
userName	String	The username for which the device is associated as default device.
installationType	String	The type of installation that is performed.
crSwitchAllowed	String	Indicates whether the device can be registered on a different Control Room.

The REST API responds to each request with an HTTP response code. For response codes, see [API response codes](#).

Parent topic: [Control Room APIs](#)