SysAid

Last Modified on 07/06/2026 9:24 am EDT

CUSTOMER ONBOARDING · SYSAID · SETUP GUIDE

This guide walks the IT/SysAid administrator through preparing a dedicated least-privilege API identity, creating a SysAid App Key, generating an access token, validating read access to the SysAid REST API endpoints that Info-Tech uses, and entering the connection details in the Info-Tech portal. The integration reads from the SysAid public REST API under `/connect/v1`.

Important — verify the authentication model and environment before relying on this guide
Two points need confirmation before publishing or relying on this setup:

1. Authentication model. Current SysAid developer documentation uses Client Credentials authentication: create an App Key that returns a Client ID and Client Secret, then use those values to generate a temporary Bearer access token. The `x-sysaid-accountid` header is required on the App Key and access-token requests. The older username/password login pattern using `/api/v1/login` is not the current documented setup flow and is not covered in this guide.

2. Confirm Spaces vs. Classic/on-premise. The steps below are written for the current SysAid public API / SysAid Spaces pattern, where examples use `https://YOUR_ACCOUNT.sysaidit.com/connect/v1`. If the customer is on SysAid Classic, a legacy tenant, or an on-premise deployment, confirm API availability and the exact URL format with SysAid before relying on this guide.

A Before you start

You will need:

- SysAid administrator permissions, or an Agent/Service Pro account with permission to create and use App Keys
- A dedicated integration Agent/Service Pro identity, preferably named `infotech-integration`
- Your SysAid **account ID**, used in the `x-sysaid-accountid` header on authentication requests
- Your SysAid account URL, typically `https://YOUR_ACCOUNT.sysaidit.com` for SysAid Spaces cloud tenants
- The **Client ID** and **Client Secret** from a SysAid App Key, or permission to create one
- A way to send test requests, such as `curl` or Postman
- Access to the Info-Tech portal where the connection details will be entered

B SysAid endpoints required

The integration requires read access to the following SysAid REST endpoints under `/connect/v1`:

REQUIRED OBJECTS

- `service-records/search` — tickets / service records, the primary source
- `activities/sr/search` — ticket activities and worklog/time data

- `agents` — agents / technicians
- `end-users` — end users / requesters
- `groups` — assignment groups and support teams

REQUIRED OBJECTS (CONT.)

- `categories` — category definitions and category hierarchy
- `companies` — company / organization reference data, if the tenant uses company values for reporting or department resolution
- `custom-fields` — custom field metadata
- `indexes` — lookup resolution for status, priority, urgency, department, and other label lookups, for example `indexes?subject=status&entity=service-record`

No service-record create, edit, delete, or write-back permissions are required for the data sync. Creating an App Key and generating an access token are authentication setup steps only.

Note: `/connect/v1/users` is not a standalone required endpoint. Current SysAid documentation exposes separate `agents` and `end-users` endpoints instead. If a tenant-specific `/users` endpoint exists in a legacy deployment, validate it separately rather than treating it as a default requirement.

Important — SysAid's permission model
Use an Agent/Service Pro identity rather than an End User. Current SysAid documentation describes Agent permissions as the control point for what a user can view, create, edit, delete, manage, and configure. End Users use the Self-Service Portal and do not have admin permissions. Configure the integration identity with the narrowest read permissions that can still access the endpoint families above.

STEPS IN THIS GUIDE

Step 1 Create the integration Agent / Service Pro

Settings → Administration / User Management → Service Pros or Administrators → New

The App Key and access token inherit the permissions of the SysAid user they are associated with. Create one dedicated, auditable integration identity rather than using a named employee's login.

1. **Create or designate the integration identity.** Use an Agent / Service Pro / Administrator-type user, not an End User. Suggested username: `infotech-integration`.
2. **Use SysAid credentials, not SSO-only credentials, if creating the App Key by API.** SysAid's

authentication guide states that App Key creation through the API requires SysAid account credentials. If the account is SSO-only, create the App Key through the UI instead or configure a compliant local credential process with SysAid.

3. **Grant the narrowest available read permissions.** The identity must be able to view service records, activities/worklog data, agents, end users, groups, categories, companies if used, custom fields, and indexes/lookups.
4. **Do not grant data write permissions unless another business process requires them.** The Info-Tech data sync does not need to create, update, resolve, delete, or write back service records.
5. **Save the account and record the username.** Store any temporary password or activation requirement in your password manager.
6. **Watch for company-based access limits.** SysAid permissions can be applied directly to an Agent or through groups. If the tenant uses company-based access limits, make sure the integration identity is not accidentally restricted away from service records or users that must be included in the reporting scope.

Step 2 Find your base URL and account ID

Your SysAid browser URL and account settings

The account ID identifies the SysAid account for authentication. The base URL identifies where the public API lives.

1. **Find the account URL.** For SysAid Spaces cloud tenants, the public API format is typically `https://YOUR_ACCOUNT.sysaidit.com/connect/v1`. Replace `YOUR_ACCOUNT` with the account-specific subdomain.
2. **Find the account ID.** In many cloud tenants, this is the same account-specific value used in the subdomain. Use the exact value expected by SysAid in the `x-sysaid-accountid` header.
3. **Record both values.** You will need the account ID for App Key and access-token generation, and the base URL for every endpoint test.
4. **If the customer is not on a Spaces cloud URL, do not guess.** Confirm the public API base URL and support status with SysAid before continuing.

Step 3 Create one App Key

Preferred UI path: Connect → Manage App Keys → Create a new App Key

An App Key creates the durable Client ID and Client Secret used to generate access tokens. Create one App Key for the Info-Tech integration and reuse it. Do not create a new key every time you need a token.

1. **Use the UI when possible.** Sign in to SysAid as the dedicated integration identity, or use a SysAdmin

workflow that creates the key for that integration identity if your tenant supports that. Go to **Connect → Manage App Keys**, click **Create a new App Key**, enter a clear name such as `Info-Tech Customer Data Store`, and continue.

2. **Save the Client ID and Client Secret immediately.** SysAid displays the Client Secret only once during initial creation. Store it in your approved password manager or secrets vault.
3. **Alternative API method.** If the UI path is not available, create the App Key with the request below. Replace the placeholders before running it:

```
curl --request POST \
  --url "https://YOUR_ACCOUNT.sysaidit.com/connect/v1/application-keys" \
  --header "Content-Type: application/json" \
  --header "x-sysaid-accountid: YOUR_ACCOUNT_ID" \
  --data '{"userName":"infotech-integration","password":"YOUR_PASSWORD","applicationName":"Info-Tech
 Customer Data Store","description":"Read-only integration for Info-Tech Customer Data Store","tokenLifet
ime":86400}'
```

4. **Expected success.** A successful App Key creation returns JSON containing `clientId`, `clientSecret`, and `applicationName`. Save the Client Secret immediately.
5. **Note the token lifetime.** The `tokenLifetime` value controls how long access tokens generated from this App Key remain valid. Use a value that matches the customer's security policy and connector refresh design — SysAid's documented default is 24 hours, with a documented maximum of 30 days.

Step 4 Generate an access token

`POST` `/connect/v1/access-tokens`

The Client ID and Client Secret are durable credentials. The Bearer access token is temporary. Use the App Key credentials to generate a token before testing or connecting the integration.

1. **Send the access-token request.** Replace `YOUR_ACCOUNT`, `YOUR_ACCOUNT_ID`, `YOUR_CLIENT_ID`, and `YOUR_CLIENT_SECRET`:

```
curl --request POST \
  --url "https://YOUR_ACCOUNT.sysaidit.com/connect/v1/access-tokens" \
  --header "Content-Type: application/json" \
  --header "x-sysaid-accountid: YOUR_ACCOUNT_ID" \
  --data '{"clientId":"YOUR_CLIENT_ID","clientSecret":"YOUR_CLIENT_SECRET"}'
```

2. **Read the response.** A successful response returns a token, token type, and expiry. Example: `{ "token": "eyJ...", "tokenType": "Bearer", "expiresIn": 86400 }`.
3. **Record the expiry.** If the connector is expected to run continuously, it must be able to generate a new token using the Client ID and Client Secret before or after the current token expires.
4. **If this fails with `401 Unauthorized`,** recheck the account ID, Client ID, and Client Secret. If the Client Secret was not saved when created, generate a new App Key and update the stored credentials.

Step 5 Verify read access to every endpoint

Run these commands from your terminal or Postman

Confirm the access token can read every API endpoint the integration needs before entering the connection details in the portal. The read calls below use the Bearer token and the account-specific base URL.

1. **Custom fields — quick credential check:**

```
curl --request GET \
  --url "https://YOUR_ACCOUNT.sysaidit.com/connect/v1/custom-fields?limit=1&offset=0" \
  --header "Authorization: Bearer YOUR_ACCESS_TOKEN" \
  --header "Accept: application/json"
```

   Success looks like JSON containing fields such as `fieldName`, `fieldCaption`, `fieldType`, and `entityType`.

2. **Service records — main ticket source:**

```
curl --request GET \
  --url "https://YOUR_ACCOUNT.sysaidit.com/connect/v1/service-records/search?limit=1&offset=0" \
  --header "Authorization: Bearer YOUR_ACCESS_TOKEN" \
  --header "Accept: application/json"
```

   Success: JSON containing at least one service record, or an empty but valid JSON result if the tenant has no matching records.

3. **Activities — worklog/time source:**

```
curl --request GET \
  --url "https://YOUR_ACCOUNT.sysaidit.com/connect/v1/activities/sr/search?hasTotalTime=true&limit=1&
offset=0" \
  --header "Authorization: Bearer YOUR_ACCESS_TOKEN" \
  --header "Accept: application/json"
```

   Success: JSON containing service-record activity records. The `activities/sr/search` endpoint requires at least one filter parameter, so this test uses `hasTotalTime=true`.

4. **Reference endpoints.** Run the same command shape against each endpoint below. Replace `ENDPOINT_PATH` with the exact path shown:

```
curl --request GET \
  --url "https://YOUR_ACCOUNT.sysaidit.com/connect/v1/ENDPOINT_PATH" \
  --header "Authorization: Bearer YOUR_ACCESS_TOKEN" \
  --header "Accept: application/json"
```

   - `agents?limit=1&offset=0`
   - `end-users?limit=1&offset=0`
   - `groups?limit=1&offset=0`
   - `companies?limit=1&offset=0` — required when company / organization labels are used in reporting

- `categories`
- `indexes?subject=status&entity=service-record`
- `indexes?subject=priority&entity=service-record`
- `indexes?subject=urgency&entity=service-record`
- `indexes?subject=department` — verify the correct entity/subject combination for the tenant if department lookup does not return data

5. **Interpret failures.** `401` usually means the access token is missing, invalid, or expired. `403` means the token is valid but the identity lacks permission for that endpoint. `400` on `activities/sr/search` usually means no valid filter parameter was provided. `404` means the base URL, path, or API availability is wrong for that tenant.
6. **When every required endpoint succeeds,** the SysAid side is ready for portal submission.

Step 6 Enter the connection details in the Info-Tech portal

Info-Tech portal

> The Client Secret and Bearer access token are credentials that give access to your SysAid data. Treat them like passwords. **DO NOT SEND THEM TO INFO-TECH.** You will also need your SysAid base URL (usually `https://YOUR_ACCOUNT.sysaidit.com/connect/v1`) and account ID.

1. Go to https://us.app.cioanalytics.ai/
2. Open the **Connections** page.
3. On the **IT Service Management (ITSM)** row, click **Connect Tool**.
4. In the **Connect to ITSM Tool** dialog, select **SysAid** from the dropdown.
5. Click **Continue**.
6. Follow the instructions in the **Setup Guide** panel on the right to fill in the fields on the left.
7. When every field is complete, click **Add Connection.**
8. Wait for the connection test to finish.

After the connection test succeeds, CIO Analytics takes over the rest of the process automatically.

1. You are returned to the **Connections** page once the connection is saved.
2. Your **SysAid** data syncs automatically for the first time. Depending on how much history is being pulled, the first sync can take anywhere from a few hours to a few days.
3. Once that first sync completes, an Info-Tech analyst will reach out to you to continue your onboarding. No further action is needed from you in the meantime.

NOTES ON THIS DOCUMENT

- SysAid navigation and exact menu paths vary by tenant and version. Use the closest matching menu

item if an exact path differs.

- This guide reflects SysAid's current public API / SysAid Spaces documentation. SysAid Classic, older tenants, or on-premise deployments may need tenant-specific confirmation with SysAid or the customer's Customer Success Manager before this guide is relied on.
- Questions about anything in this guide can be directed to your Info-Tech onboarding contact.
