Turbine User Guide

1. Quickstart

1.1. Quickstart Overview

Welcome to the Turbine User Guide Quickstart section! This section helps you get started with Turbine quickly and efficiently.

What's in Quickstart?

The Quickstart section provides everything you need to begin using Turbine:

Getting Started

- **Getting Started Basics** - Supported browsers, login methods, navigation, and key terms
- **Set Up Your Profile** - Configure your user profile and preferences
- **Best Way to Start**- Recommended learning path and best practices

Turbine Cloud

- **Turbine Cloud** - Cloud-specific features, security, and tenant management
- **Security-Specific Features** - Database security and account security guidance

Try a Solution

- **AI SOC Solution** - A complete, ready-to-use security operations workflow that demonstrates Turbine's capabilities

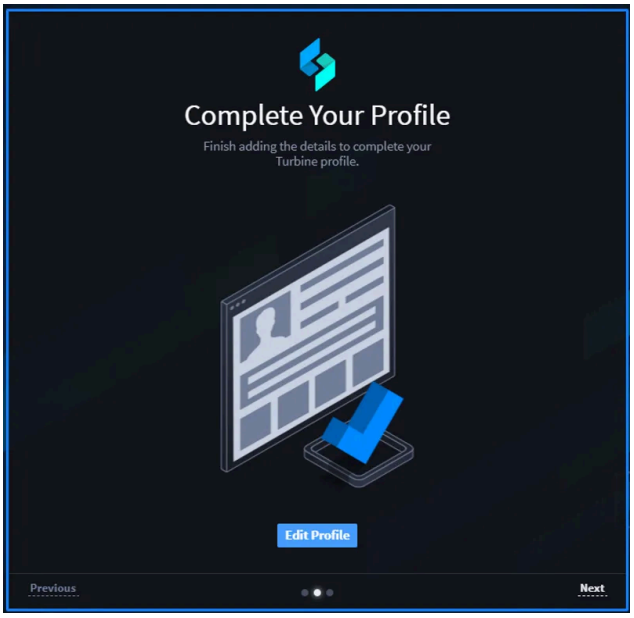
What's Next?

1.1.4.1. Customize Your User Profile

Swimlane Turbine provides flexibility in managing your user profile, allowing you to customize personal settings and manage access tokens, roles, and groups.

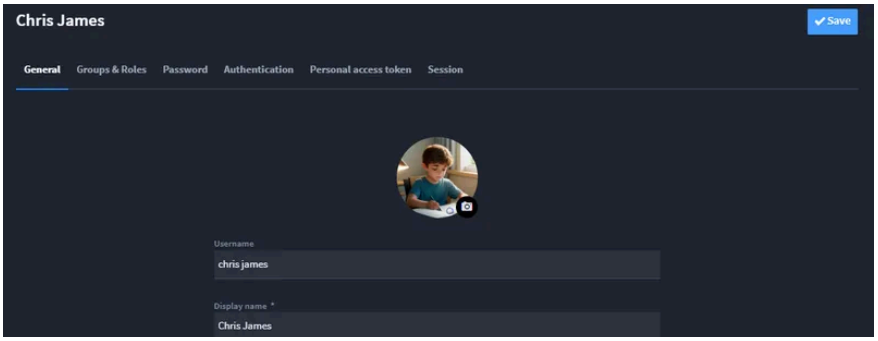
Completing Your Profile

Upon your first login, you'll see the **Complete Your Profile** screen. Follow these steps to finalize your profile:



This action opens the **User Profile Editor**, allowing you to:

- Upload a profile picture to personalize your account.
- View the account's most recent activity.
- Update general details such as your display name, email, and time zone.
- Assign groups & roles to control permissions (Admin only).
- Enable or disable user accounts (Admin only).



2.2. Daily Operations Overview

Daily Operations covers all the tasks you perform regularly in Turbine to view, manage, and work with your data.

What's in Daily Operations?

Workspaces

Organize your workspace and navigate between different views and dashboards.

Key Topics:

- Navigate Workspaces and Dashboards
- Workspace management

Dashboards

Create and manage dashboards with charts, visualizations, and interactive elements.

Key Topics:

- Creating and managing dashboards
- Dashboard features (charts, colors, sorting, date ranges)
- Dashboard permissions and sharing
- Dashboard filtering options

Application Records

Work with records in your applications - search, filter, edit, and manage data.

Key Topics:

- **Working with Records** - Search, filter, color code, lock, restrict records
- **Bulk Operations** - Bulk modify, bulk restrict and lock

2.2.3.5. Deleting or Editing a User Dashboard

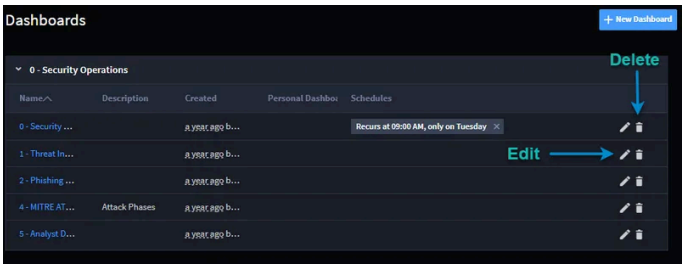
To delete a dashboard:

From the Dashboards taskbar menu, select **Delete**.

You can change the settings of the dashboard by clicking on **Settings and Schedules**. The Dashboard Settings and Schedules pages allows you to edit the following:

- **General Settings**: Edit the **Name**, **Description**, and **Workspaces**.
- **Advanced Settings:** Change the timeline filter duration.
- **Timeline Filters:** Select date fields to apply global date range filters across applications.
- **Schedules:** Create or edit the schedules.
- **Permissions:** Edit the permissions.

You can also edit or delete a dashboard from the Dashboards page. Click the pencil (edit) or the trash can (delete) icon.



2.2.3.6. Set Dashboard Permissions

To modify the dashboard permissions, from the Create or Edit Dashboard dialog, click the PERMISSIONS tab. Dashboards can be accessible personally or through role-based access control.

You can also set up private dashboards. Private dashboards allow end-users to create personal dashboards that only they can view and manage.

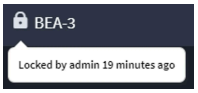
If you have permission, you can also set up private dashboards. A private dashboard can only be viewed or modified by whoever created the dashboard. Administrators or the creator of

2.2.4.2.1. Record Lock

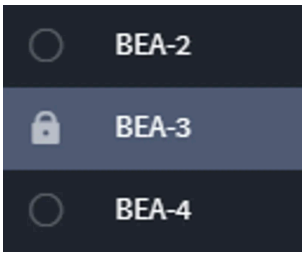
You can lock records while editing and modifying them in order to prevent your changes being overwritten by another user. In addition, locked records can not be bulk modified or bulk deleted.

Once a record is locked, the lock icon appears next to the record's Tracking ID in the Record header.

When a record is locked, only the user who created the lock or an administrator can unlock it. To unlock a record, from the Record header, access the record menu and select **Unlock Record.**

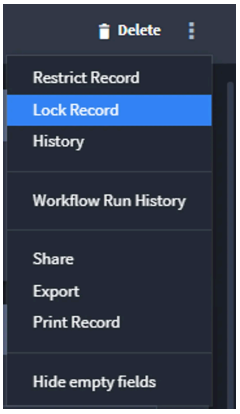


Locked records are also indicated in the report-view-of-records (Default Report) interface.



To lock a record:

1. Open a record. From the record taskbar, access the record menu.



2. From the menu, select **Lock Record**.
   The record is locked immediately. Anyone can access the individual record and open it, but only the user who created the lock or an administrator can make changes to the

2.2.4.5. Advanced

2.2.4.5.1. Lookup and Create References within Records

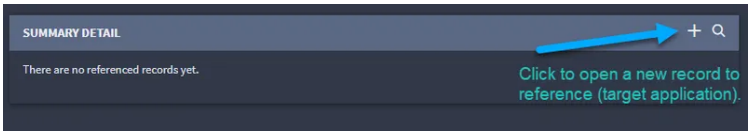
You can select which field(s) to reference from within a record. A reference field on a record, whether single-select, multi-select, or grid, is accompanied by a New and Lookup button adjacent to the reference field.

For more information about reference fields in general, see **Reference**.

Creating a New Record to Reference

To create a new record to reference:

1. From within an open record, locate the reference field, and then click **+**, or Add New. A record editor for the target application opens.



2. Fill out the target record data.

Looking Up References within Records

To lookup and create references within records:

1. Click within the field, or click the Lookup (Search) icon. Enter a keyword or search term and then press Enter to begin the search.

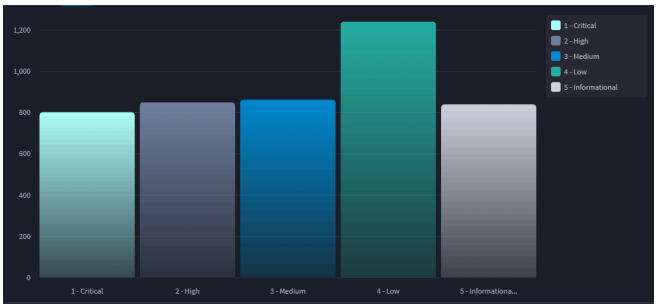
2.2.5.4.2. Chart Types

This topic contains chart type examples.

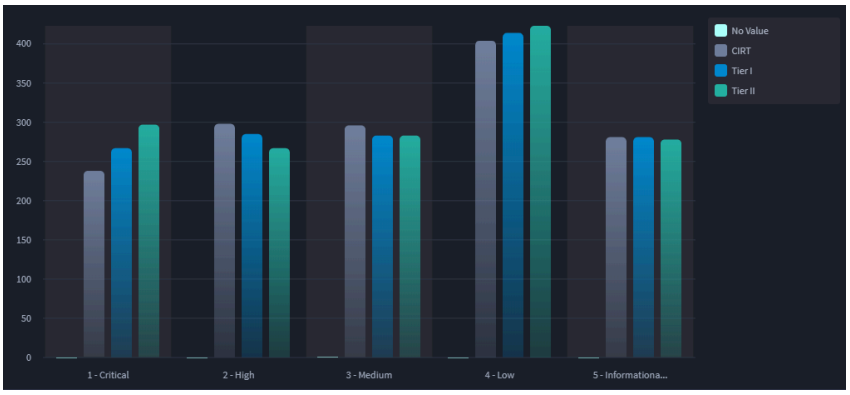
Bar Charts

Use bar charts to show comparisons between different categories of data. The bars can display either vertically or horizontally.

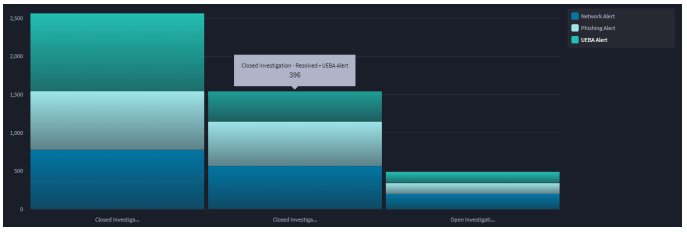
Vertical Bar



Grouped Vertical Bar



Stacked Vertical Bar



100% Vertical Bar

2.3.1.1. Getting Started

2.3.1.1.1. Playbooks Overview

Playbooks are a series of triggers, logic, and actions that automate a workflow. A playbook can contain triggers, actions, native actions, components, assets, inputs, and outputs.

> To create or change flows with natural language, use **Playbook Building Mode** (Text to Playbook). See **Create and Modify Playbooks with Hero AI** or the **Hero AI** overview.

Playbook Architecture

- A playbook can have one or more **flows**.
- Each flow has exactly one trigger.
- Flows do not communicate with each other.

For details, see [**Flows**](../03-Using Actions/03-Flows/02-02-69-Flows.md).

Playbook Data Model

On the canvas, Turbine stores playbook information in **two layers**. Understanding both layers explains what **Save** updates and why some changes appear only after you save from the canvas editor.

Layer	What you see in the UI	What is stored	What runs
Playbook	Playbook name, description, canvas	A **builder playbook** record that lists which flows belong	Does not run by itself; groups flows

2.3.1.3.1.4. Schedule Triggers

Schedule triggers initiate a playbook at a configured time using cron expressions. They are ideal for automating repetitive tasks, improving productivity, and reducing human errors. For example, you can use a schedule trigger to automatically generate reports, eliminate the need to manually retrieve data, or perform periodic maintenance tasks.

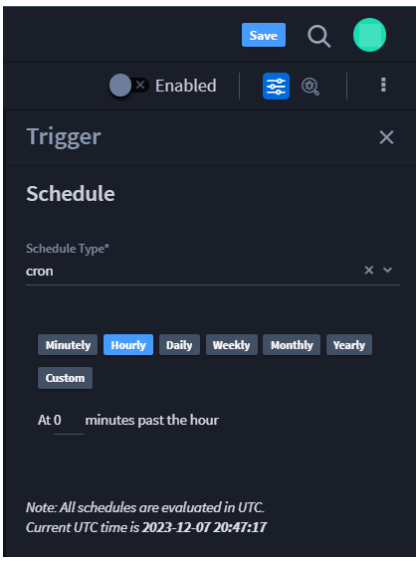
Key Benefits

- **Automated Execution:** Run playbooks automatically at specified intervals without manual intervention
- **Flexible Scheduling:** Support for various schedule frequencies (minutely, hourly, daily, weekly, monthly, yearly, or custom cron expressions)
- **UTC Timezone:** All schedules are evaluated in UTC for consistency across different server locations
- **Reliable:** Uses Hangfire job scheduler for reliable execution of scheduled tasks

Creating a Schedule Trigger

To set a cron job for your playbook, follow these steps:

1. In a playbook, from the Add panel, click and drag **Schedule** to the canvas.
2. Hover over the plus icon to add it to the canvas. The Trigger panel will display on the right side of the canvas, where you can configure your schedule trigger.



2.3.1.4.4.1. Create Record

The **Create Record** native action creates a new record in a selected application from the playbook builder. The action maps data to application fields and returns the tracking ID of the created record.

Overview

The Create Record action enables you to programmatically create records in Swimlane applications from within playbooks. This is useful for automating record creation based on events, data from other systems, or calculated values from upstream actions.

Key Benefits

- **Native Action:** Integrated directly into the playbook builder for streamlined workflow creation.
- **Flexible Field Mapping:** Map data from upstream actions, playbook inputs, or expressions to application fields.
- **Record Restrictions:** Optionally restrict record access to specific users or groups.
- **Tracking ID Output:** Returns the tracking ID of the created record for use in downstream actions.
- **At Most Once Execution:** Ensures the record is created exactly once, even if the playbook is retried.

Inputs

The Create Record action requires the following inputs:

Required Inputs

- **Application:** Select an application by either:
  - **Application ID**: The unique identifier of the application
  - **Application Name**: The name of the application
- **Note:** You must provide either `applicationId` or `applicationName`, but not necessarily both.

2.3.1.4.10. Using the Parallel Native Action

The **Parallel** native action in Swimlane Turbine allows orchestrators to execute multiple paths in parallel, waiting for all branches to complete before continuing downstream. This action is ideal for workflows requiring simultaneous processing of tasks with synchronized results.

Overview

The Parallel native action enables simultaneous execution of multiple actions, optimizing workflows that require concurrent processing. By ensuring all parallel tasks complete before proceeding, the action provides precise control over complex playbook flows.

Key Benefits of the Parallel Action

- **Native Action:** Integrated directly into the playbook builder for streamlined workflow creation.
- **Concurrent Processing:** Enables simultaneous execution of multiple actions, reducing overall processing time.
- **Synchronized Results:** Ensures that all branches complete before the workflow continues, maintaining workflow consistency.
- **No Connector Needed:** Operates within Swimlane without external integration.
- **WAIT Feature:** Provides a visual indicator showing that the playbook is waiting for parallel branches to complete before proceeding.

Understanding the Parallel Action

Using the Parallel native action, you can execute multiple paths simultaneously, wait for them to complete, and then continue the playbook flow. The action provides **On Success**, **On Failure**, and **On Complete** paths based on the outcome of the parallel branches.

How Parallel Actions Work

1. **Entrypoints:** All entrypoint actions defined in the Parallel group are executed simultaneously when the Parallel action starts.
2. **Conditional Entrypoints:** Entrypoint actions can have conditional `if` expressions. If

2.3.1.7.1. Inputs

Actions are individual capabilities of connectors that interact with external technologies by passing in data via action inputs. Results are available from action outputs.

Actions can:

- Call another playbook
- Interact with external technologies
- Contain inputs and outputs (outputs can be promoted through the **Playbook Outputs** dialog)

> You do **not** need a trigger to add an action to your playbook.

Create Actions

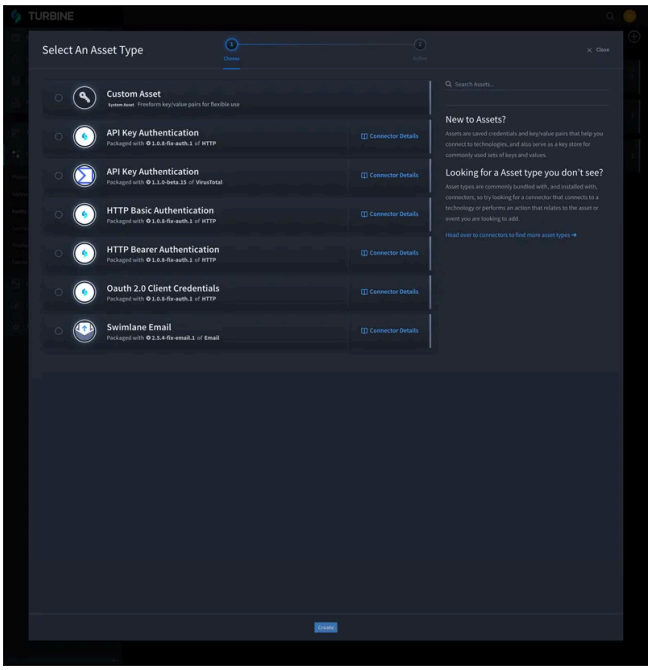
1. From your current playbook, open the **Add** panel on the left side of the screen. The **Add** panel displays available actions, connectors, and components organized by category. The panel has three tabs: **Triggers**, **Actions**, and **Components**.
2. Browse or search for the desired action from the available connectors and native actions.
    - Native actions (such as Create Variables, Update Variables, HTTP Request, Script, Transform Data, Condition, Loop, Parallel, Create Record, Delete Record, Search Records) appear at the bottom of the **Add** panel
    - Connector actions are organized by vendor/connector above the native actions
3. Drag the action from the **Add** panel.
4. Drop the action onto the canvas:
    - Drop onto a drop zone (indicated by a plus icon) to add it to the flow
    - Drop onto an edge to insert it between existing actions
    - Drop onto a node to connect it after that action
5. After dropping, the action appears on the canvas and the **Action** details panel automatically displays to the right of the playbook.
6. In the **Action** details panel, under the **Info** section:

2.3.2.3.1. Create Custom Assets

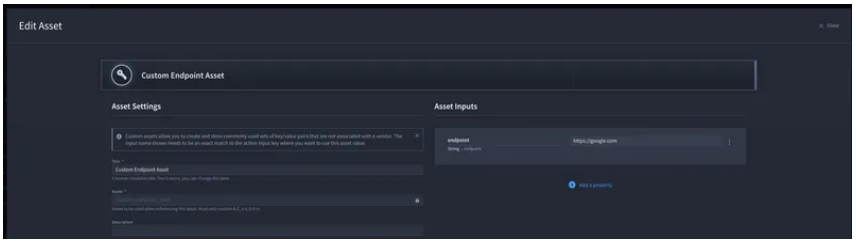
Creating custom assets allows you to map static data and product-specific values that can be referenced across multiple playbooks and multiple actions.

To create assets:

1. On **ORCHESTRATION**, click **Assets**.
2. Click the plus icon to add and/or configure an asset.
3. On **Select An Asset Type**, click the radio button next to the **Custom Asset** connector.



4. On **Edit Asset**, update the required fields with custom data: **Title** and **Name**.
5. Click **Add a property** to add asset inputs.



6. Click **Save**.

Apply Custom Assets in Playbooks

1. You can apply a custom asset to a playbook input

2.3.4.1.1. Context Variables

Context variables are variables that hold a variety of contextual information relating to the current playbook, its invocation parameters, and executed actions.

Using context variables is especially helpful when creating **Playbook Triggers** to help define parameters and retrieve/store specific data.

This table lists some of the helpful context variables you can use.

Context Variables	Description	Example
$playbook	The name of the playbook	find_iocs

$inputs	User-defined schema of inputs for a playbook	"$inputs": {
		"some_string_playbook_input": "some_string_passed_to_the_playbook"
		}

$event	Data related to the sensor event that triggered the playbook	**Webhook**
		
		"$event": {
		"data": {
		"headers": {
		"x-account-id": "228edce1-be3e-4b45-99d5-b4a11ac41489",
		"x-tenant-id": "911b9bae-540e-4064-93dc-33c1364587dd",
		"host": "turbine_webhook_agent_pool",
		"connection": "close",
		"authorization": "Basic U3dpbWxhbmU6YXNk",
		"user-agent": "PostmanRuntime/7.28.3",
		"accept": "*/*",
		"cache-control": "no-cache",

2.3.4.1.6. Nested Playbooks

You can nest playbooks into other playbooks as actions. Refer to **Actions** for instructions to add an action to a playbook.

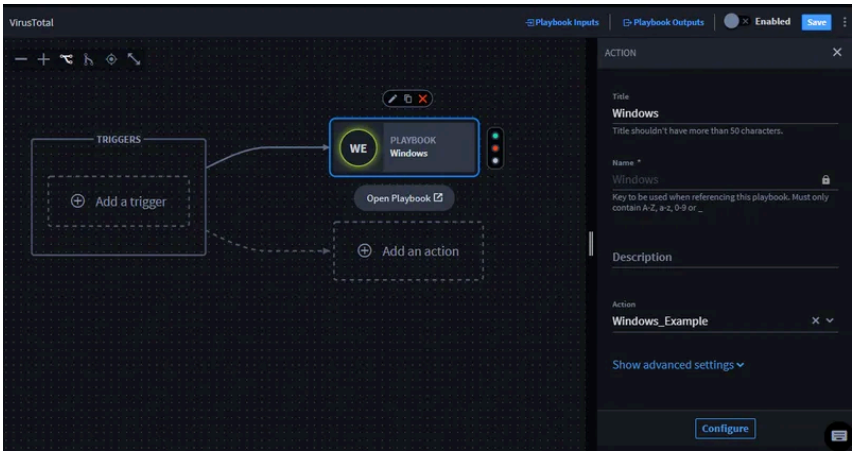
Terms

Term	Meaning
Parent playbook	A playbook that calls another playbook.
Nested playbook	A playbook that calls another playbook as an action.

> When nesting a playbook into a parent playbook, the nested playbook's inputs and outputs are available to map to and use in the parent playbook.

To nest a playbook to a new/current playbook:

1. Navigate to the **Playbooks** home page, and then open a playbook (or create/upload a playbook).
2. Click **Add an action**.
3. From **ACTION**, click the **Action** drop-down menu, and select an existing playbook.
4. From **ACTION**, add a title.



5. Click **Configure** to map inputs and outputs.
6. If the nested playbook has inputs and/or outputs, they are available on **Action Inputs**.
7. Review **Action Outputs**.

2.3.4.1.9. Classic Native Actions

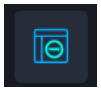
Native actions are actions that you do not need to add manually and that provide a customized experience for easy configuration and optimal performance in Turbine.

This section covers features including native actions and python chat.

Native Actions are:

- Located at the bottom of the Add panel
- Turbine's collection of native logic utilities or logic actions

Native Actions in the UI:

Icon	Meaning
	Condition
	Create Variable
	HTTP Request
	Loop
	Parallel
	Script
	Transform Data
	Create Record
	Delete Record
	Search Record
	Upsert aka Update/Create Record

2.3.4.1.9.1. HTTP Request

2.3.4.1.9.6. Transform Data

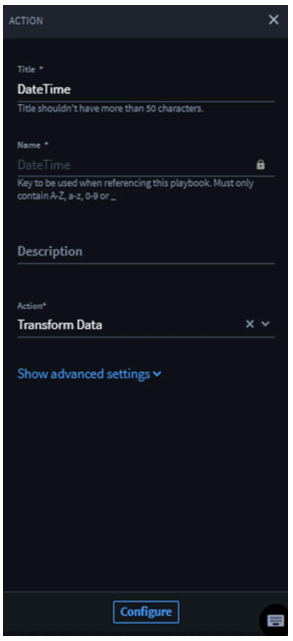
One of the most common data transformation jobs is converting data to specific formats. Historically, this operation has required Python or JSONata expertise. Turbine eliminates this need by using a transformation action in a playbook.

Basic Transformation Action Set Up

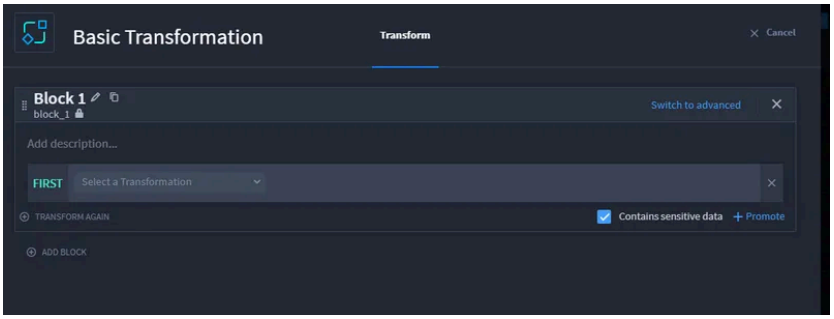
Time to start basic set-up for the Transform Data action.

You have already created a playbook, and you are ready to transform the data from a property.

1. From your playbook, click **Add an action**.
2. From the **ACTION** panel, click the **Action** drop-down.
3. Select **Transform Data** and then click **Configure**.



The transformation builder opens.



2.3.4.1.11. Playbook File Handling

You can use a connector to handle files in playbooks by using a file-handling connector in an action. For example, you can use a connector that ingests an email that contains attachments.

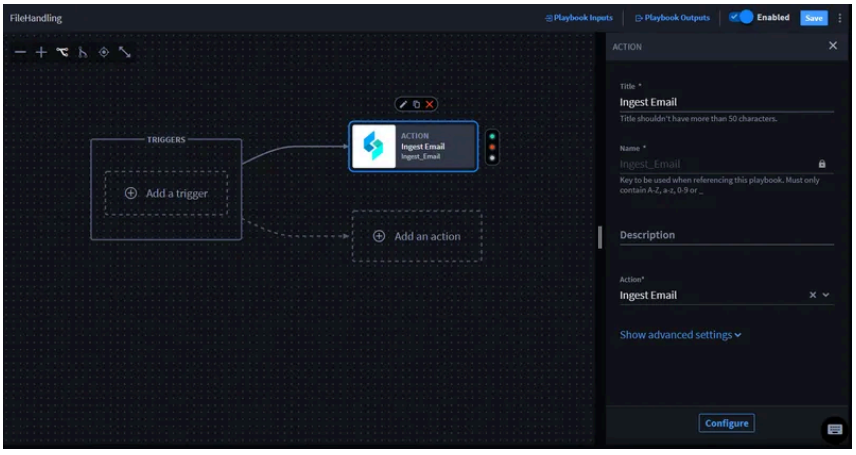
The file is an attachment from an action output.

> There is a limit of 500MiB per file.

Use Files with a File-Handling Connector

To add files with a file-handling connector:

1. Open a playbook (or create/upload a playbook).
2. Click **Add an action**.
3. From **ACTION**, add a title.
4. From the **Action** drop-down menu, select a connector that has the capability to handle files.



5. Click **Configure** to map the property values.
   The action window displays with the **Inputs** and **Outputs** tabs.
6. Click **Outputs**.
   For a playbook to pass a file as data, you need to add a file as an action output.
7. Click **+ Promote** next to the attachments.

2.4.1.1. Applications and Applets

Applications and Applets are the foundation of the Swimlane Turbine platform.

What is an application?

A user-defined template for collecting, storing, and organizing your data. All automated activities and decisions are driven by how your application stores data. You also manage workflow from within applications.

What is an applet?

A preconfigured set of fields and layout specifications. Applets are appended to an existing application form layout and allow users to easily update and expand their existing applications.

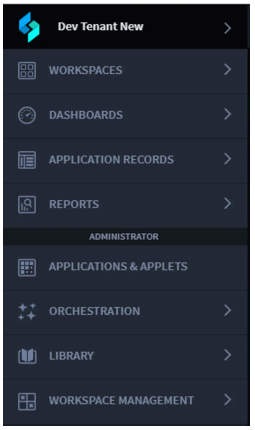
This section lists all the Applications and Applets that have been created and are available on this Turbine instance. You can filter the list by type (Application and Applets), or by a search string that you enter.

Each Application and Applet provides a management menu with viewing options, copy and export options, and a delete option.

To review or reconfigure the application, click **Builder**.

2.4.1.2. Manage Applications and Applets

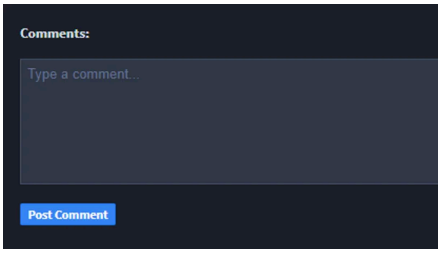
The Applications and Applets home page is where you manage your applications and applets.



2.4.2.6.2. Comments

Use this field to create a comments section in the record.

Here is how a comments field appears within a record:



To create a comments field:

From Application Builder's Field Types, select **Comments** and then drag and drop it to the Form Layout. Drop the field in the layout area, or within a tab or section layout object.

Access the field's properties tab and complete the following fields as needed:

Field	Step	Example
Display Name	Enter the name of the field.	*Comments*
Help Text	Enter contextual help text. You will first need to specify whether the help text will appear above or below the field in the record form, and then you can enter the text.	*Attach related files here.*

Add specific field-level permissions by role, and when ready click **Apply.**

2.4.2.6.3. Correlation

You can correlate records within a single application within Swimlane Turbine. Upon ingestion of new application records, Turbine compares the new record to previous records that have correlation keys.

Record Correlation executes two tasks:

2.4.3. Building Applets

2.4.3.1. Applet Builder

Applet Builder is the starting point for building and managing applets. With Swimlane Turbine's Applet Builder you can:

- Create and modify fields and layouts.
- Set field level permissions and properties.
- Configure calculations.
- Access, build, and update workflow.
- Run manual validations.
- Create export templates.
- View revision history.
- Delete applets.

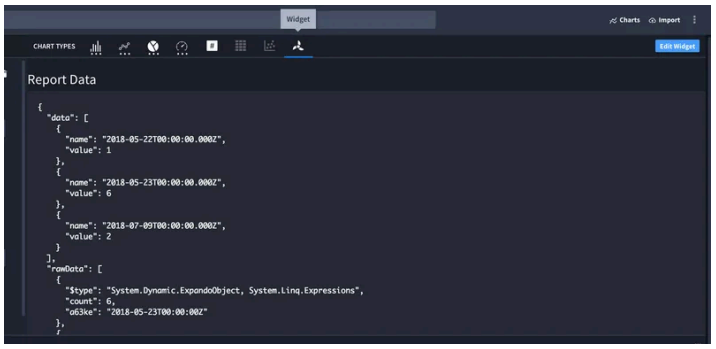
Applets share nearly all of the same features as applications. There are a few important differences, including the way applets handle reference fields and the way that applets configure workflow upon the addition of an applet to an application.

Note: Users who do not have permissions set on any applications, but have global permissions set in all other areas of Turbine, will be able to view applications and applets in the Administrator menu. If users have global permissions for applets, they will be able to edit applets. However, since they do not have any permissions set for applications, they will see applications listed, but will be unable to edit them or otherwise modify them.

Page Components

2.4.4.4. Report Widgets

You create report widgets from within the Charts feature on the Default Reports page. Click **Charts** from the Default Reports taskbar, then from **Chart Types,** select the Widgets icon.



Report widgets have access to the report's data, in two different formats: raw (as it is received from the API), and a more user-friendly format, where the ID's are replaced with values. In addition to the report's data, the widget also has access to the query.

Example data (in json):

```
{
  "data": [
    {
      "name": "Domain",
      "value": 1
    },
    {
      "name": "Email Address",
      "value": 5
    },
    {
      "name": "File Hash",
      "value": 1
    },
    {
      "name": "IP Address",
      "value": 2
    }
  ],
  "rawData": [
    {
      "$type": "System.Dynamic.ExpandoObject, System.Linq.Expressions",
      "count": 1,
      "aAPdorKWs_pxNlGkJ": "5acba8b218d0f6f1c9b89b5a"
```

2.4.6. Calculations

2.4.6.1. Calculation Builder

Fields specified as calculations are read-only. The value of a calculated field is the result of a calculation defined with a custom formula. Calculated fields can be either text or numeric, but cannot be lists.

Calculated Fields are processed on all Record GET (including the default Record for App endpoint), Record POST and Record PUT in addition to the explicit Calculate Field endpoint ("app//record//calc/"). Swimlane Turbine uses Synfusion Calc to process calculations, and also uses the same syntax. The success and/or failure of calculations is recorded in logged activity in Turbine.

> Be aware of the output type of your calculation when creating the calculation result field.

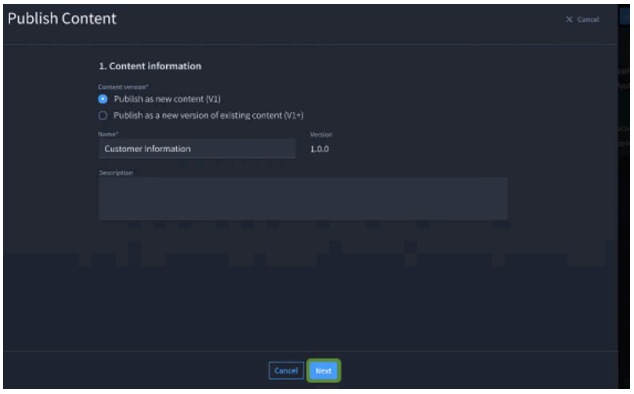
The Calculation Builder consists of the following:

- Formula editor window
- Formula operator buttons
- Validate formula button
- Expression Helper
- Formula functions
- Help Text

2.4.7.1. Publishing an Applet

An Applet can be published to the Turbine Library as content that can be shared across your Turbine account. The publishing process includes:

1. **Publish as New Content (V1)**
2. **Publish as a New Version of Existing Content (V1+)** (including handling merge conflicts if a remote repository has been configured)



Publish as New Content (V1)

This process applies when the applet is being published for the first time.

Steps:

1. Open the applet settings menu and select **Publish Applet** .
2. In the **Publish Content** window:
3. Select **Publish as New Content (V1)**.
4. Provide a name and description for the content.
5. Click **Next**.
6. Select Content to Publish:
    - All the applet components (workspaces, playbooks, reports, etc.) will be listed. Check the components to be included in the package.
    - Click Next.
7. Review Content:
    - Review the selected content and check for potential issues.
    - If everything looks good, click **Next**.

2.5.3.6. Collaborating with Swimlane Users

Swimlane Turbine enables you to securely collaborate with Swimlane support staff without requiring them to be onboarded as internal users. This gives you the flexibility to grant or revoke access as needed, without worrying about licensing limits, authentication management, or persistent user accounts.

Accessing External User Settings

1. Go to the **Admin Panel**.
2. Select **Settings** from the sidebar.
3. Click the **Account** tab.
4. Open the **External Users** sub-tab.

Allow External Users

You can allow external Swimlane users to access your account by enabling the **Allow existing external users** toggle. These users must already exist in the Swimlane account to be eligible for access. **Click Save after you enable it.**

- **When the toggle is ON:**
  - External users can be invited and added to your account.
  - Previously disabled external users will be reactivated, provided they are still active in their original Swimlane environment.
- **When the toggle is OFF:**
  - All external users associated with your account are immediately disabled and cannot log in.
  - New external users cannot be invited until you turn the toggle back ON.

After enabling the toggle, click **Save** to display the **Invite external users** and **View all users** buttons.

Inviting External Users

You can invite external Swimlane users to your account in two ways:

2.5.4.1.3. Directory Services

Swimlane Turbine integrates with two Directory Service types: Microsoft's Active Directory (AD) and Open LDAP. By integrating with Directory Services, administrators can streamline user management and ensure consistent authentication across their organization.

Use Cases and Benefits

Many Turbine administrators leverage Directory Services to:

- Enable SOC Engineers and Analysts to log in to Turbine with previously established directory credentials.
- Automate user and group management, reducing administrative overhead for large teams.
- Increase security by centralizing authentication and maintaining compliance with corporate policies.

> Users are synced upon each login. Automatic synchronization occurs every night at midnight, server time.

These settings are at the account level and propagate to all the tenants associated with users through roles or groups.

Enabling Directory Services

> Before you begin, verify that your server settings are correct and ensure you have the necessary permissions to configure Directory Services.

To enable Directory Services:

1. Click on your profile, and then click **Admin Panel**.
2. From the left navigation, select **Settings** > **Account**.
3. Click **Directory Services** and then select **Enable Directory Syncing**.

This expands the selections for the settings that you need to configure.

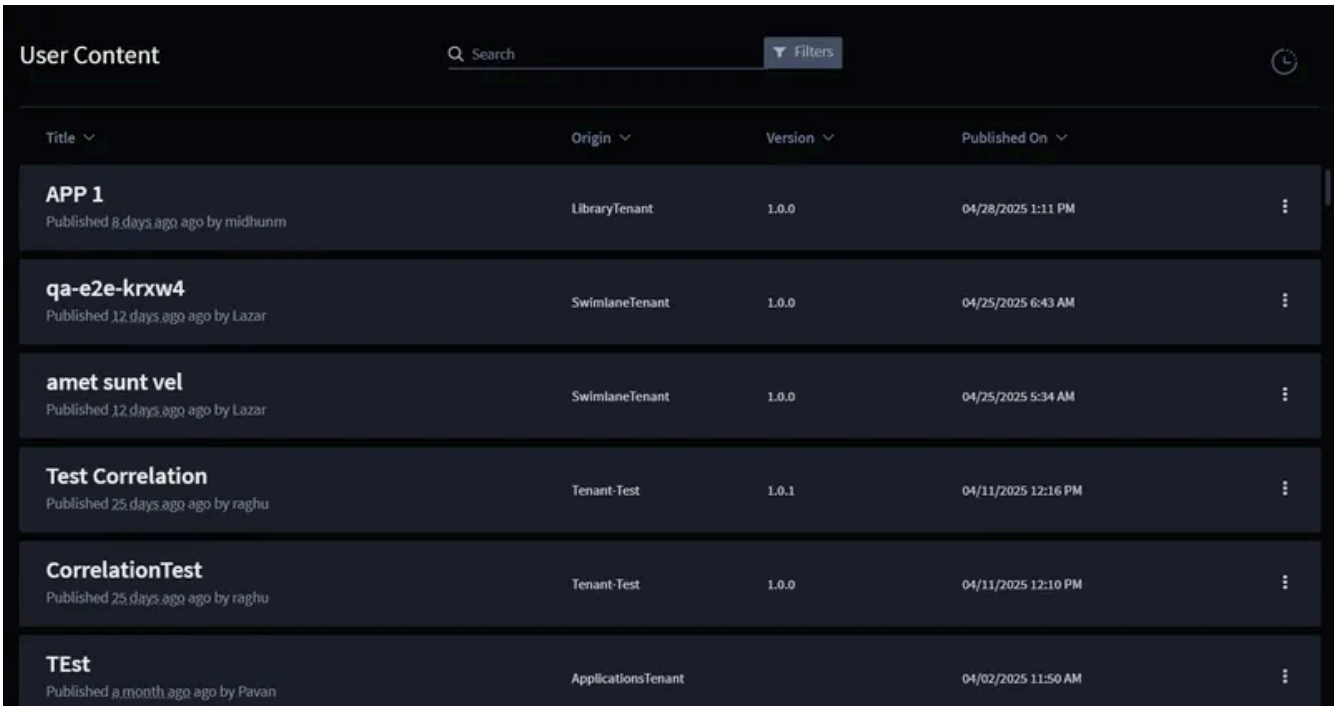
2.5.6. Content Management

2.5.6.1. Library - Admin Panel

Library section from the **Admin Panel** lists all the content packages that are published to the **User Content** page of the **LIBRARY** from any of the Tenants on the Account and the **Configuration Manager**. This page allows you to manage the content package, deploy from one tenant to another tenant, and manage assets

2.5.6.1.1. User Content Homepage

The homepage displays a list of content packages, along with the ability to Push and Pull changes with Git integration if a remote repository is configured.



2.5.7.1.4. Notifications Dashboard

The Notification Dashboard helps in tracking the number of notifications generated and emails sent across tenants and playbooks. It provides visibility into notification activity, email distribution, and user-level email metrics within the selected filters.

The following metrics are displayed:

- **Total Notifications Generated:** Displays the total number of notifications generated for the selected date range, tenants, playbooks, and actions.
- **Total Emails Sent:** Represents the total number of email notifications sent within the selected range.

Tabs

The dashboard contains two tabs:

- **Tenants**
- **Emails Generated**

Each tab provides a different level of visibility into notification activity.

Tenants Tab

The **Tenants** tab provides an overview of notification activity across tenants via a line graph and a detailed metric table.

Graph

A line graph displays the notifications generated over time for the selected date range:

- The horizontal axis represents the selected date range.
- The vertical axis represents the number of notifications generated.
- Hovering over a data point displays the total notifications generated for that specific date.

Table Details

2.5.8.4. Troubleshooting

2.5.8.4.1. Troubleshooting and Diagnostics

Collecting Logs

- Use the logs command mentioned in **Reference: Validation and Commands** section to review and collect logs for support.

Network Test

Validate connectivity to Swimlane Turbine Cloud:

```
docker run --rm quay.io/swimlane/turbine-agent:<version> network-test https://<region>.sw
```

Replace `<version>` with your agent version and `<region>` with your Swimlane Cloud region.

> This test does not work with privately hosted Swimlane Turbine instances. For the test to work, you must allow outgoing connections to `*.amazonaws.com`.

2.5.8.4.2. Reference: Validation and Commands

2.6.1.2. SOC Interfaces

This document lists the interface contracts available in the SOC Solutions Bundle. For general information about what interfaces are and how to use them, see **Working with Interfaces**. For complete data model field definitions, see **Turbine Schema Reference (Classic SOC)**.

SOC Solutions Bundle Interfaces

The SOC Solutions Bundle includes 20 interfaces for Security Operations Center workflows. Use these interfaces for alert triage, observable enrichment, email processing, and remediation actions.

Alert to None v1.0.2

Purpose: Processes an alert object without producing output. Use this interface for alert ingestion workflows where you process alerts without transforming them.

Input schema:

Field	Type	Required	Description
`alert`	object	Yes	Alert object
`alert.alert_categories`	array of strings	No	Alert categories
`alert.alert_created_timestamp`	string	No	When the alert was created
`alert.alert_description`	string	No	Alert description
`alert.alert_end_timestamp`	string	No	Alert end time
`alert.alert_impacted_hostnames`	array of strings	No	Affected hostnames
`alert.alert_impacted_ip_addresses`	array of strings	No	Affected IP addresses
`alert.alert_impacted_usernames`	array of strings	No	Affected usernames

3.3. Create and Modify Components with Hero AI

Create and Modify Components with Hero AI

Use **Component Building Mode** to create or change a component on the canvas with natural language. Hero's **component builder agent** applies your prompts to the component you have open in the Component Builder.

This mode is different from **How Hero AI Executes Components**, where the **general companion** finds and runs components marked **Visible to Hero AI** from chat outside the builder.

For how Hero AI switches between companion and building modes, see **Hero AI Companion**.

Choose Your Path

I want to…	Start here
Build or change a component on the canvas with AI	**Enter Component Building Mode**
Run an existing component from chat (not edit the canvas)	**How Hero AI Executes Components**
Make a component discoverable to the companion	**Components** — **Visible to Hero AI**
Understand companion vs building modes	**Hero AI Companion**

Enter Component Building Mode

There is **no separate Building Mode toggle** in settings. The companion enters **Component Building Mode** when Hero AI opens **with component canvas context** (the product knows which component you are editing in the Component Builder).

How you open Hero AI	Mode	Typical use
Hero AI from the main application toolbar while you are **not** in the Component Builder	General companion	Questions and chat not tied to a component canvas; see **Hero AI Companion**

4.1.1. Accessibility Conformance Report

- **Name of Product/Version:** Swimlane Turbine Cloud 25.3.0
- **Report Date:** October 2025
- **Product Description:** Swimlane Turbine is a low-code security automation platform. With Turbine you can prioritize alerts, re-mediate threats and improve your operational performance.
- **Contact Information:** info@swimlane.com

> Turbine is a web-only application.

This report covers the degree of conformance for the following accessibility standard/guidelines:

Standard/Guideline	Included in Report
Web Content Accessibility Guidelines 2.2	Level A (Yes) Level AA (Yes) Level AAA (Yes)

Terms

The terms used in the Conformance Level information are defined as follows:

- **Supports:** The functionality of the product has at least one method that meets the criterion without known defects or meets with equivalent facilitation.
- **Partially Supports:** Some functionality of the product does not meet the criterion.
- **Does Not Support:** The majority of product functionality does not meet the criterion.
- **Not Applicable:** The criterion is not relevant to the product.

Success Criteria, Level A

Criteria	Conformance level	Remarks and explanations
1.1.1 / Non-	Parti	Some images that contain text are missing `alt` tags. Some icon

4.2.9. Monitoring Script for Remote-Agent Host Health

This article provides a script that customers can use to collect system health metrics from the server running the Turbine remote-agent containers. The goal is to capture key data points (DNS resolution, CPU, memory, I/O, network performance, etc.) at regular intervals so that, in the event of a container restart or connectivity issue, system conditions at the time can be reviewed.

This is particularly useful in environments where connectivity to the Turbine Cloud instance may be impacted by DNS timeouts, network quality issues, or resource exhaustion.

This solution has been tested on commonly used Linux distributions, including:

- **RHEL 7 / 8 / 9**
- **Ubuntu 18.04 / 20.04 / 22.04**

What the script collects

- Running containers (docker ps)
- Container IPs
- Uptime
- Memory usage
- CPU load (top processes)
- Disk I/O (iostat)
- Disk usage
- DNS resolution
- Network response times using curl

Prerequisites

- Docker/Podman must be installed and running
- Optional but recommended: iostat via sysstat package
  Install the sysstat package:

4.3.1.3. Configure String Array Condition Expressions

Turbine allows users to configure conditions between two actions by utilizing playbook inputs and/or action outputs.

Scenario

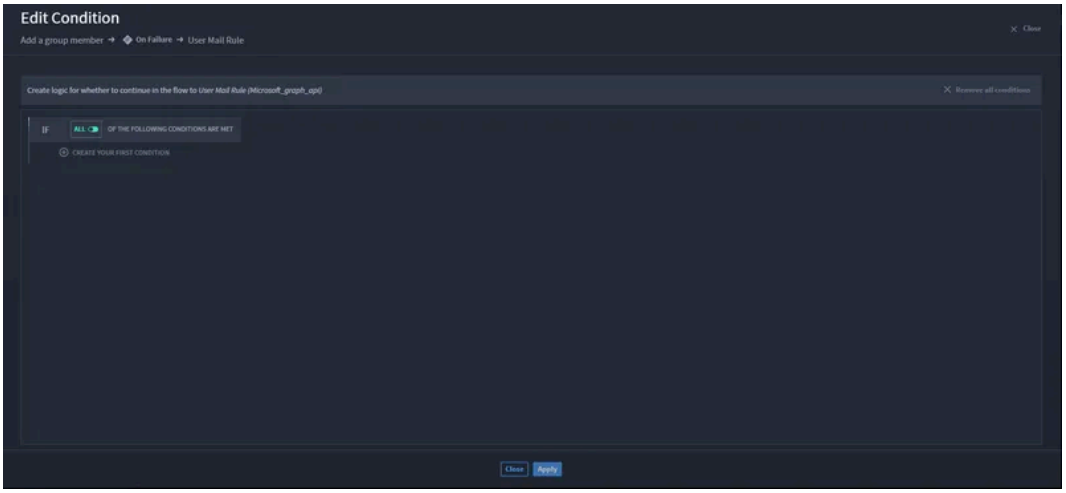
Alex wants to configure a string array in a conditional expression using the output of a JSONata action. Alex is ready to begin. He starts by adding and configuring a JSONata connector with an **On Success** action flow, and now he wants to add a string array conditional expression.

To add a condition, click the **Add condition** icon.

> Once you click the action flow, it turns blue and the FLOW panel displays to the right

The Edit Condition window opens.

Click **CREATE YOUR FIRST CONDITION**.



The available action/playbook properties displays. Click the property to expand the available input types.

Click **String Array** property.

> If available, you can select more than one String Array.

4.3.3.3. Action Retries

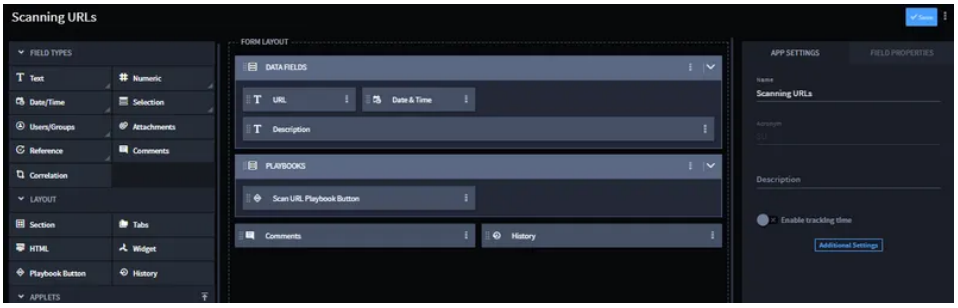
Scenario

Alex works in a Security Operations Center (SOC). She is responsible for discovering and reporting malicious URLs. She receives a report of URLs each morning. She looks up each of the URLs, and then uses the resulting scans to report which URLs are most severe. This takes up most of Alex's morning each work day.

Alex wants to use Turbine to simplify and automate this process. She has already ensured that her version of Turbine has downloaded and configured the Urlscan connector from Swimlane Content and created/saved a Scan URLs playbook, and reviewed the report to determine which data she wants to use to feed her Swimlane application(s) and playbook.

Alex can resolve this scenario with the following solution:

Build an application that includes data fields and a playbook button that serves as a record action trigger.



Alex created the **Scanning URLs** application with the following data fields: **URL**, **Date & Time**, and **Description**, and added the **Playbook Button** to link her **Scan URLs** playbook that she already created.

With the application complete, Alex can return to the **Scan URLs** playbook to configure her data.

> The playbook trigger is the Record Action (via the application playbook button).

Add the **Submit** connector that will submit URLs for analysis.
Alex is ready to configure the connector by adding the URLscan asset, which she preconfigured with the needed URL and API Key data.

4.3.6.2. Webhook Discovered Outputs

Scenario

Eli is an orchestrator who wants to test and preview outputs from a webhook event before configuring and saving the rest of his playbook. Luckily, as an orchestrator, Eli knows that testing a webhook event is like testing **discovered output and testing**. Let's watch how he tests his webhook event!

Eli has already created his playbook and added a Webhook trigger. After giving the webhook a name and generating the URL, he's ready to configure.

1. Click **Configure**.
   The Webhook Events, Outputs, and Map to Playbook Inputs tabs are available. Eli checks the Output tabs to see the original outputs for the webhook. But he wants to test for any additional outputs that might not be being pulled. To do this, he needs to test the webhook.
2. Click **Test** next to the URL and view the Result pane at the bottom of the window.
   The results display, but Eli wants to know which outputs were discovered that were not part of the original event outputs. Since Eli already reviewed and understands what **discovered output and testing** are, he is ready to add the discovered outputs.
3. Click the check mark next to any discovered outputs to add them to the action's outputs.

Conclusion

These are now available for Eli to apply downstream and as he builds the playbook.