Data Connectors Integration Guide

This guide explains how to create and use custom data connectors to integrate external systems with ZoomInfo.

Before you begin **2**
- What is a data connector? 2
- When should I use a connector? 2
- Benefits and outcomes 3
- Key terms 3

Create a new data connector **4**
- Set up the authentication method 6
- Connect your data connector 8
- Troubleshooting 9

Create data connector Actions **9**
- Before you create an Action 10
- Create an Action 10
- Customize the Action Form 18

Use a Connector Action in an Audience **24**

Before you begin

This guide is intended for administrators and technical users who have permission to manage integrations and access to their external system's API documentation and credentials.

Confirm that the external system provides an API and that you have the required credentials, permissions, and API documentation. Creating a connector does not require custom code in ZoomInfo, but it does require configuration details from the external system.

Confirm your ZoomInfo permissions

- Administrator access to GTM Studio or the Admin Portal.
- Permission to create and manage integrations.

Gather external-system API details

- Base API URL
- Authentication method and credentials, such as OAuth credentials or an API key
- Required headers, beyond authentication
- Sample API request and response JSON
- Expected input types and required fields for the endpoints you plan to use

Confirm the authentication method

Custom data connectors support the following authentication methods:

- OAuth 2.0
- API Key

What is a data connector?

- A connector defines how ZoomInfo securely connects to an external API.
- An action defines a specific API operation available through that connector.
- A form determines how a user provides or maps action inputs in ZoomInfo.

When should I use a connector?

Before creating a custom data connector, check whether ZoomInfo offers a prebuilt integration for the external system and scenario you need. To find available prebuilt integrations, from the main navigation, go to **Integrations > Connections > Integrations**.

Use a custom data connector when a prebuilt integration is not available or does not support your required scenario. A custom connector lets ZoomInfo interact with the external system's API.

You can use a connector to:

- Retrieve external information to enrich an Audience.
- Send ZoomInfo data to an external system.
- Create or update records in another application.
- Trigger an operational process in an internal system.

Benefits and outcomes

Custom data connectors let you integrate ZoomInfo with external systems that have supported API access.

Data connectors provide the following benefits:

- **Extend your technology stack:** Use data and capabilities from the external systems your organization relies on within ZoomInfo.
- **Control your data flow:** Configure the data ZoomInfo sends to and receives from each external system.
- **Automate repeatable work:** Retrieve data, enrich Audiences, create or update external records, and trigger downstream processes.
- **Reduce custom development:** Configure supported API interactions in ZoomInfo without building a one-off code integration. You still need the external system's API details and appropriate credentials.

Key terms

Term	Description
Data connector	A reusable setup that tells ZoomInfo how to communicate with an external system, including the system's base URL and authentication method.
Connection	An authorized link between ZoomInfo and a specific account in that external system. A connection is created when a user completes OAuth authorization or provides an API key.
Action	A reusable API call, such as retrieving a record, enriching data, creating a record, or updating an external system.
Parameters	Inputs used in an endpoint URL, request body, headers, or a data-source action.
Output schema	The fields ZoomInfo exposes from an Action's response for use in later steps.

Form	The fields shown to ZoomInfo users when they run an Action. A form lets users enter values, select options, or map available ZoomInfo fields to the inputs required by the external API.
Dynamic dropdown	A dropdown whose values are retrieved from another Action on the same connector.

Example we'll use throughout this guide:

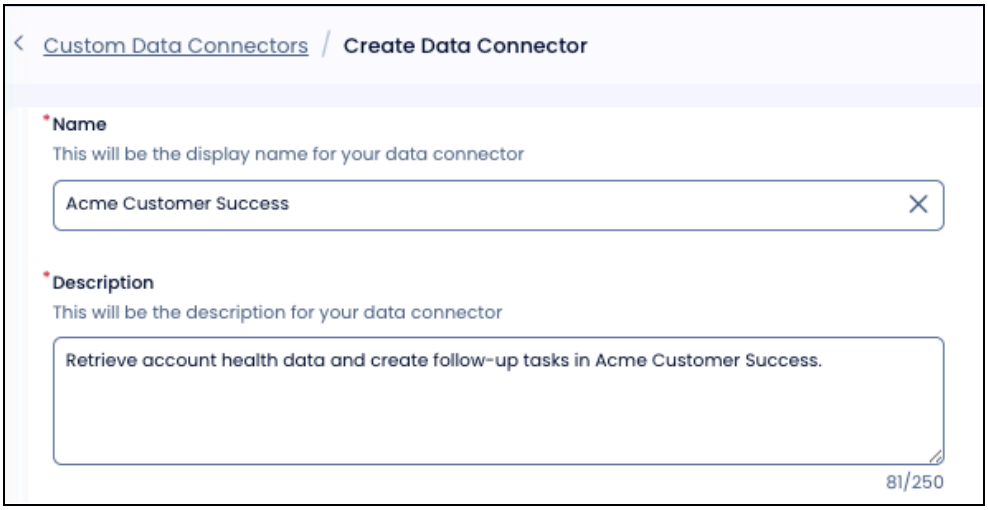
Your team uses a customer-success platform, ACME Customer Success, that is not available as a prebuilt ZoomInfo integration.

1. Create a **connector** that defines how ZoomInfo connects to the platform.
2. Create a **Get account health** Action that retrieves account health data by company domain.
3. Configure a **Form** that maps the company domain from your ZoomInfo Audience to the Action.
4. Use the Action to return account-health fields, such as health score and renewal date, as Audience enrichment data.

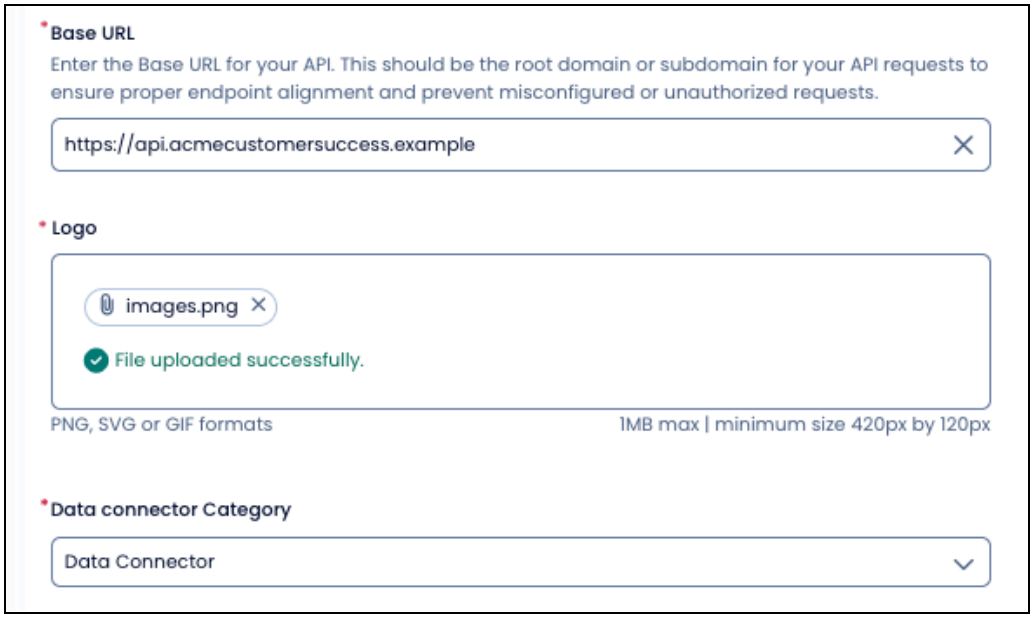
Create a new data connector

This step creates the connector configuration; it does **not** create the API Actions. The connector can have multiple Actions using the same authentication configuration.

1. Navigate to **GTM Studio/Admin Portal**.
2. Select **Integrations** from the navigation menu.
3. Select **Custom Data Connectors**.
4. Click **Create data connector**.
5. Complete the required fields:
    - **Name**: Use the vendor/system name that your users will recognize.
    - **Description**: State what users can accomplish with the connector.



- **Base URL**: The base URL to access the API for the connector. Avoid adding versions (i.e. /v2) as these can change and be endpoint-specific (endpoint paths are added in a later step).
- **Logo**: Image for the data connector. PNG, SVG or GIF formats only (1MB max).
- **Data Connector Category**: Choose based on the intended business use.



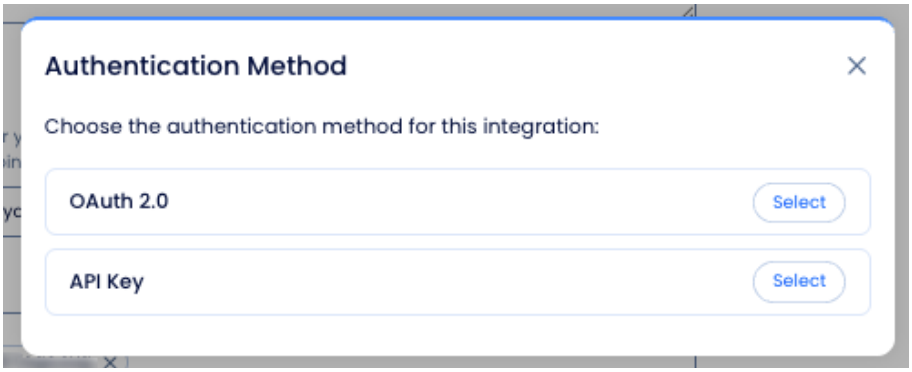
6. Under **Authentication method**, select **Set Up**. Then move on to the next section.
7. **Calls per second:** Optionally set the maximum number of API requests ZoomInfo sends to the external system per second.
    - Use this setting when the external system has documented API rate limits or when you need to limit load on the destination.

- If you do not set a calls-per-second limit, ZoomInfo sends requests as quickly as available processing capacity allows. For high-volume Actions, this can cause the external system to return rate-limit errors, such as HTTP `429 Too Many Requests`.
- Check the external system's API documentation to determine an appropriate limit.

Set up the authentication method

Authentication is configured at the connector level and reused by its Actions. ZoomInfo's custom data connectors support vendors that use OAuth 2.0 and API Key authentication methods.

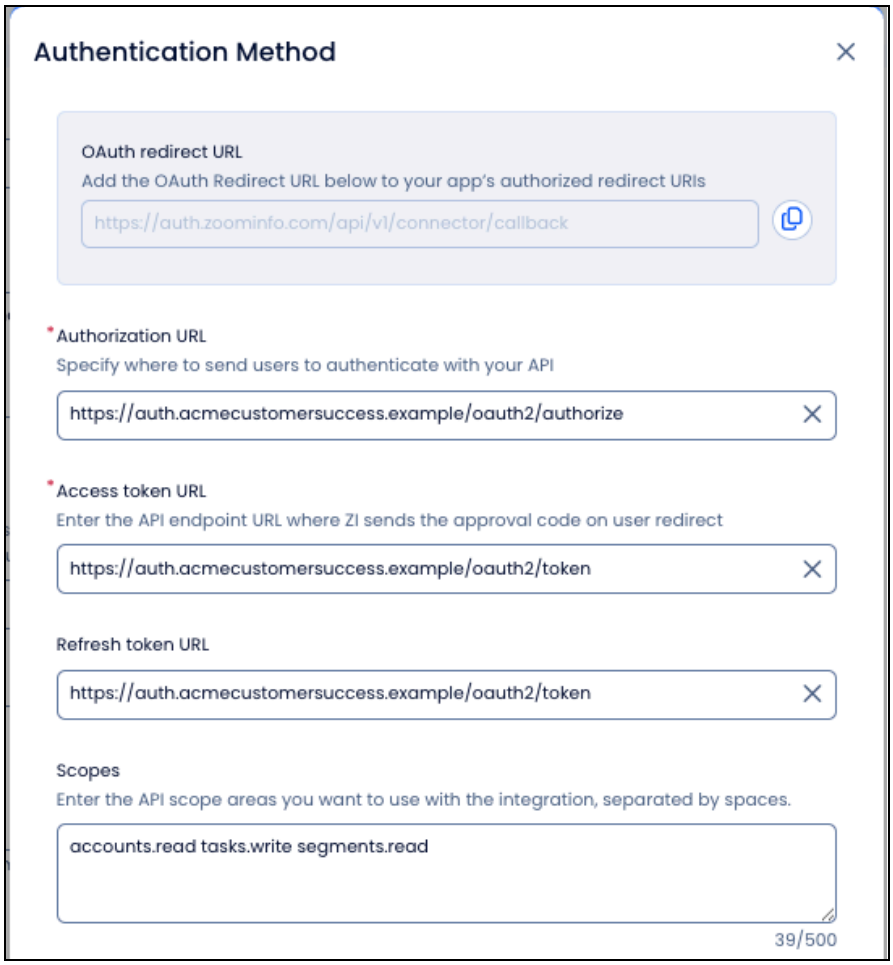
1. Select the authentication method based on your external system.



OAuth 2.0 authentication

If using OAuth 2.0 as your authentication method, complete the following fields:

1. **OAuth redirect URL**: Copy this URL and add it to your connected application's authorized redirect URIs.
2. **Authorization URL**: Enter the Authorization URL (e.g., https://auth.example.com/oauth2/authorize) that specifies where to send users to authenticate with your connected app's API.
3. **Access token URL**: Enter the Access Token URL (e.g., https://auth.example.com/oauth2/token). This is the API endpoint URL where ZoomInfo will send the approval code on user redirect.
4. **Refresh token URL**: Enter the endpoint ZoomInfo uses to exchange a refresh token for a new access token, if required by the external system (e.g., https://auth.example.com/oauth2/refresh_token).
5. **Scopes**: Enter the permission required by the external API, separated by spaces.



6. **Client ID**: Enter the Client ID.
7. **Client secret**: Enter the Client Secret.



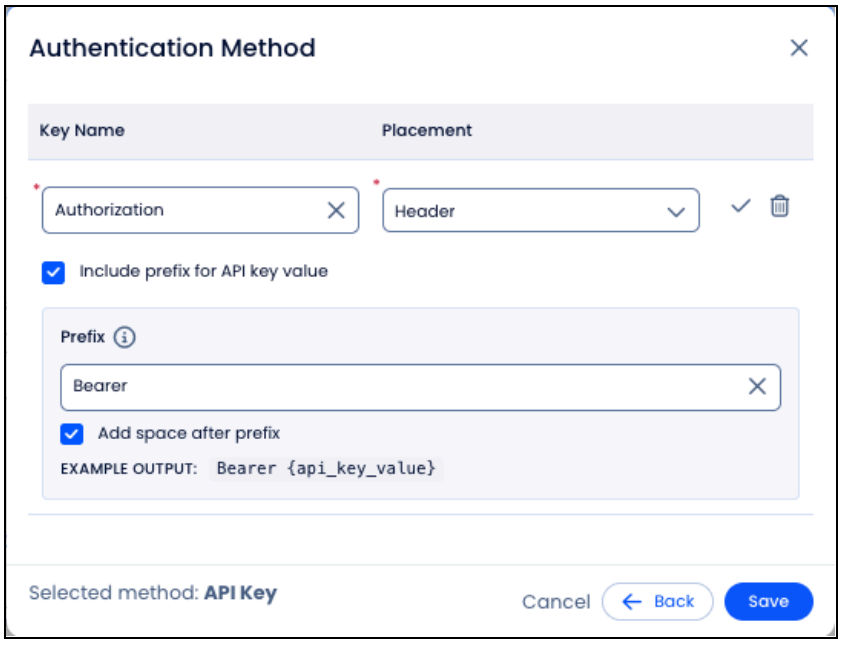
8. When done, click **Save**.

API Key authentication

Use **API Key** authentication when the external system requires an API key or token with each API request.

1. Select **API Key** as the authentication method.

2. Enter the following authentication details:
    - **Key Name**: Enter the parameter or header name required by the external API. For example, enter `Authorization` or `X-API-Key`. Do not enter the API key value in this field.
    - **Placement**: Select where the external API expects ZoomInfo to send the API key (Query parameter, Header, or Body)
3. If the external API requires text before the API key, select **Include prefix for API key value**.
    - In **Prefix**, enter the required text, such as Bearer.
    - Select **Add space after prefix** when the API requires a space between the prefix and API key.
4. Review the **Example Output** to confirm that ZoomInfo will format the API key as required by the external API.
5. Select **Save**.

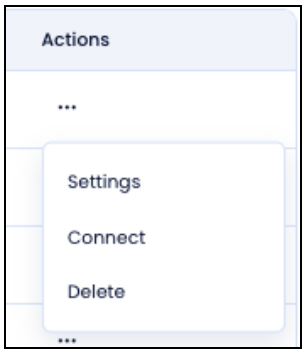


When a user connects the data connector, they enter the API key value. ZoomInfo sends the key using the configured key name, placement, and optional prefix.

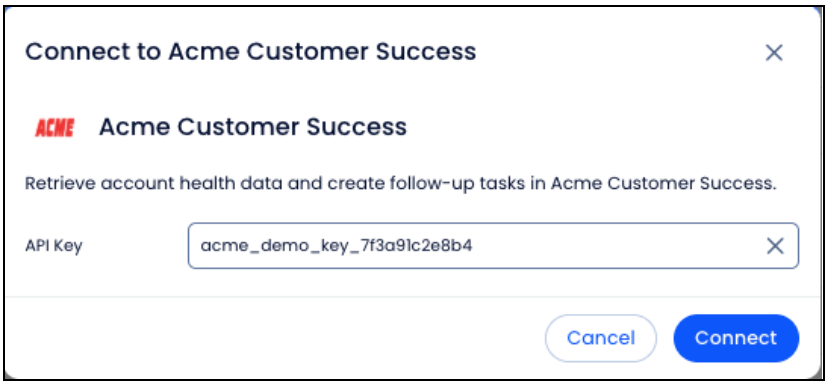
Connect your data connector

Once the data connector is configured, it can be connected from the **Custom Data Connector** list view.

1. Select the menu under Actions > **Connect**.



2. Complete the authentication process.
    - If using **OAuth 2.0** as your authentication method, you'll be redirected to the external system to authorize. Complete the authorization process on the external system.
    - If using **API Key** as your authentication, enter your API key and click **Connect**.



Troubleshooting

1. Confirm that the base URL and credentials are correct.
2. Ensure required API-key headers/prefixes are configured.
3. Verify the Oauth redirect URI and scopes with the vendor.
4. Confirm the user/account has permission to call the intended API endpoints.

Create data connector Actions

An Action is a reusable API request. For example, you can create Actions to retrieve account details, enrich Audience records with external data, create a record in another application, or update an existing record.

After you create and connect a data connector, create Actions to interact with specific endpoints in the external system's API.

Each Action uses the authentication configuration defined for its connector.

Before you create an Action

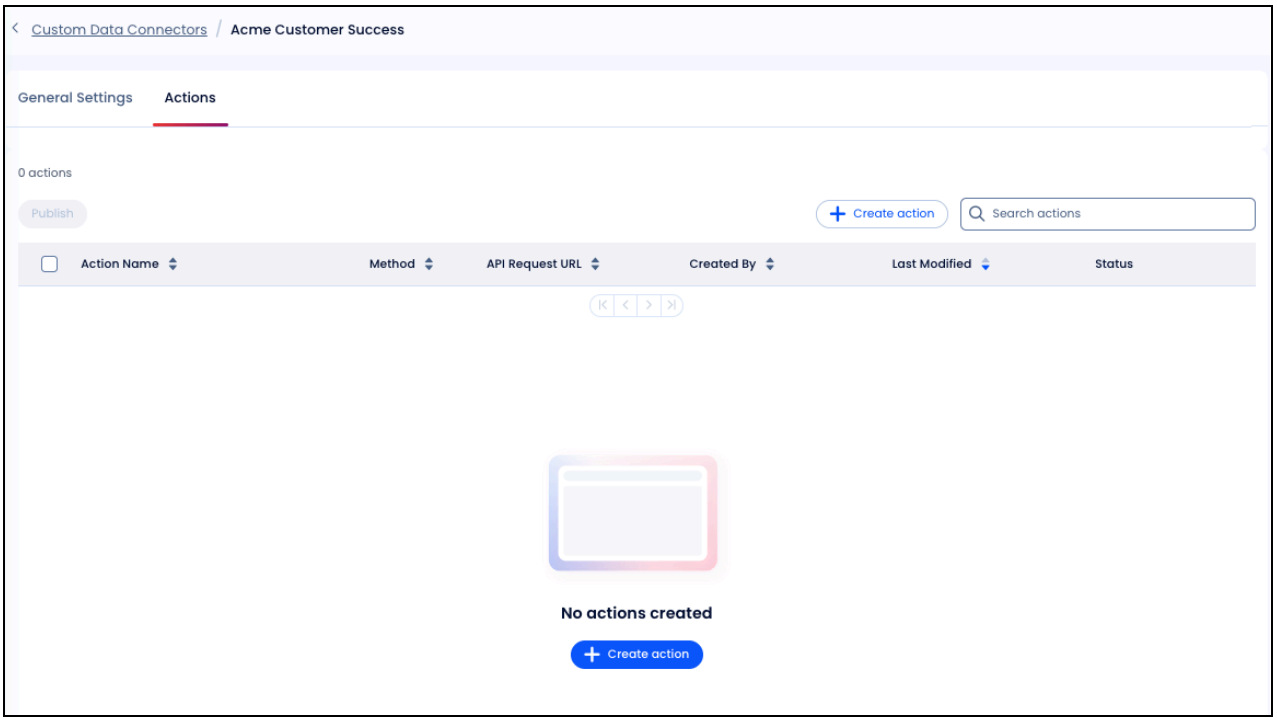
Identify the API endpoint you want to call and gather the following from the external system's API documentation:

1. The endpoint path and HTTP request type
2. Required URL parameters, headers, and request-body fields
3. Sample request and response JSON
4. The response fields that ZoomInfo users or downstream steps need

Create an Action

1. From the **Data Connectors** page, select the connector you want to configure.
2. Select the **Actions** tab.

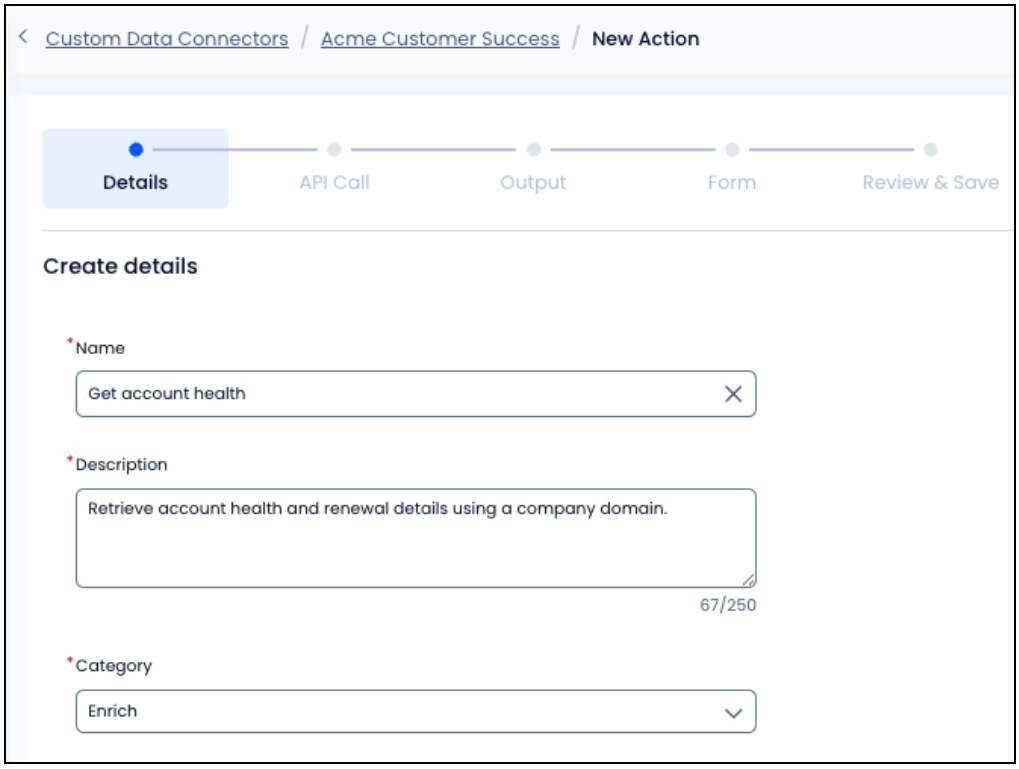
   Here, you can create one or more Actions for the connector. Each Action corresponds to a single API call (e.g., GET <data>, PUT <data>, and so on).
3. Select **Create action**.



Each connector can include one or more Actions. For example, a connector could include one Action to retrieve account data and another Action to create a record in the same external system.

Step 1: Enter Action details

1. On the **Details** step, complete the following fields:
    - **Name**: A clear, outcome-oriented name that users will recognize, such as *Get Account Details* or *Create Contact*.
    - **Description**: A brief explanation of what the Action does.
    - **Category**: Select the Action's business purpose:
        - **Retrieve:** Get information from the external system.
        - **Enrich:** Return external data for use as enrichment in ZoomInfo.
        - **Export:** Send ZoomInfo data to the external system, such as to create or update a record.



The action Category describes the Action's business purpose. The request type describes the HTTP method ZoomInfo uses to call the external API. For example, an Export action might use a POST, PUT, or PATCH request.

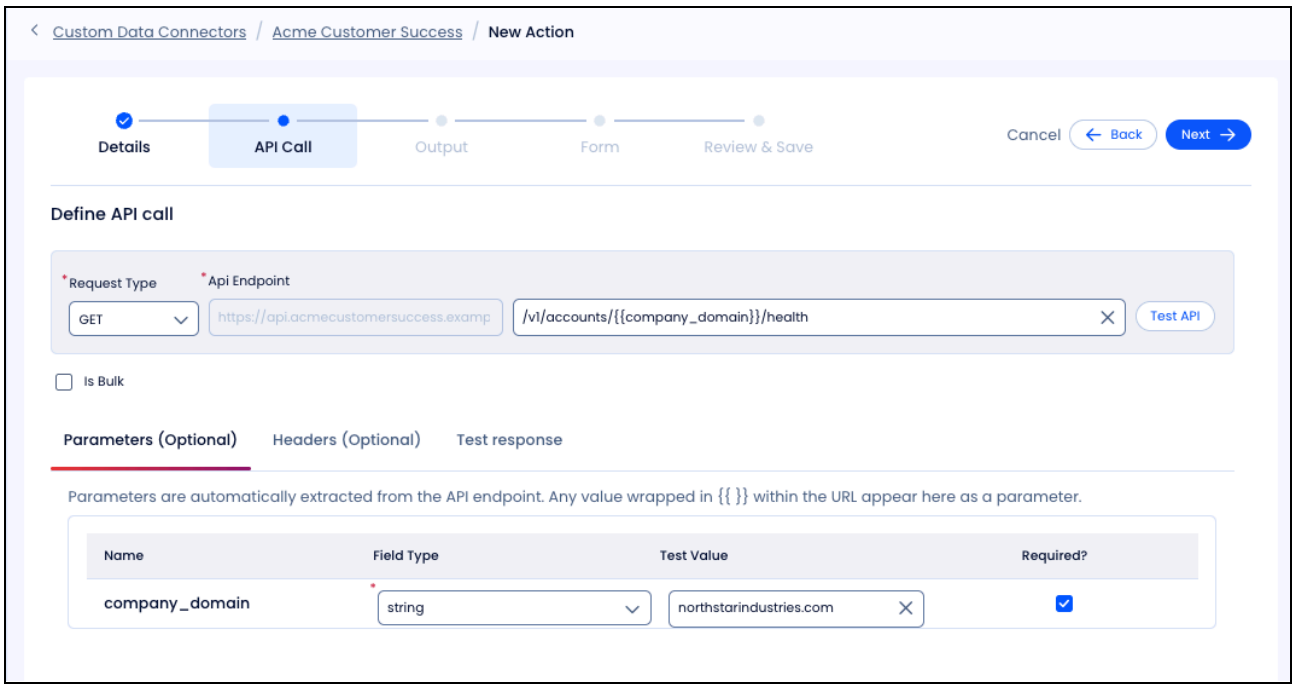
2. Select **Next**.

Step 2: Configure the API call

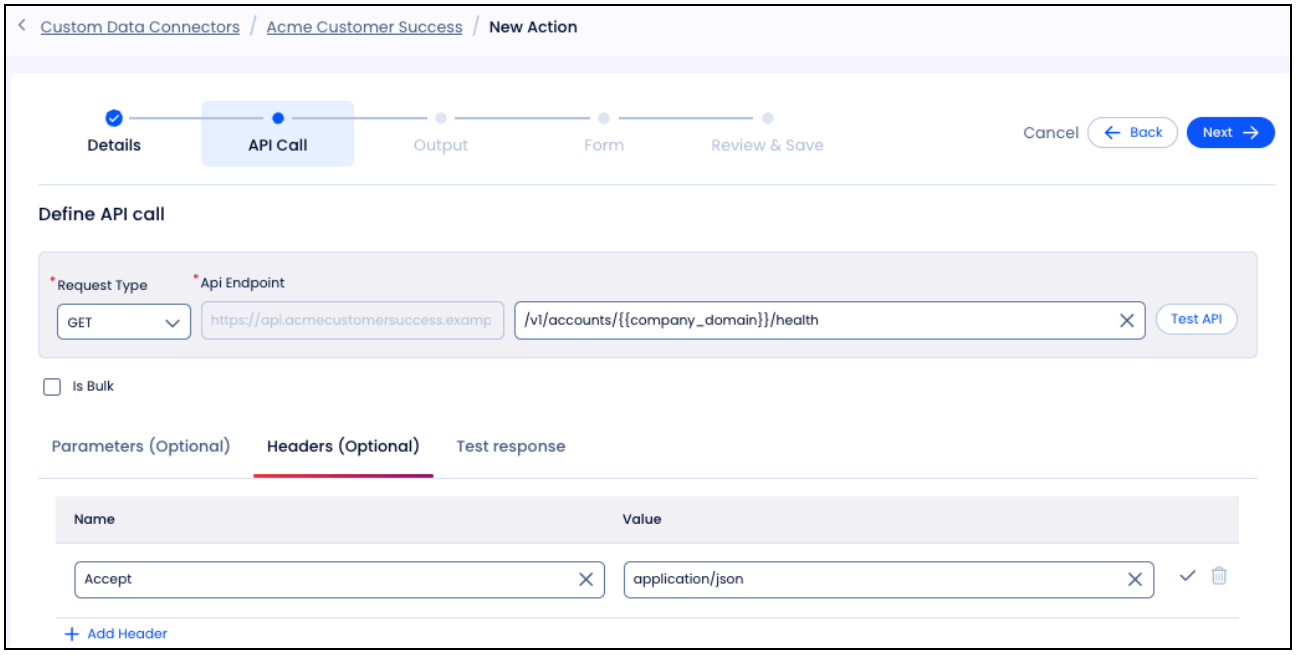
On the **API Call** step, configure how ZoomInfo calls the external API.

1. Select the **Request Type** (GET, POST, PATCH, PUT, or DELETE)

2. Enter the **API Endpoint URL**.
    - Add any inputs that should be provided dynamically by using the `{{parameter}}` format in the endpoint URL.
    - ZoomInfo automatically identifies values enclosed in double braces and adds them to the **Parameters (Optional)** tab.
3. Configure **Parameters (Optional)**
    - Use clear, descriptive parameter names that reflect the external API's requirements.
    - Select the appropriate field type: **string**, **number**, **date**, **boolean**, or **array**.
    - Mark the parameter as **required** when the Action cannot run without it.
    - Add test values before testing the API call.



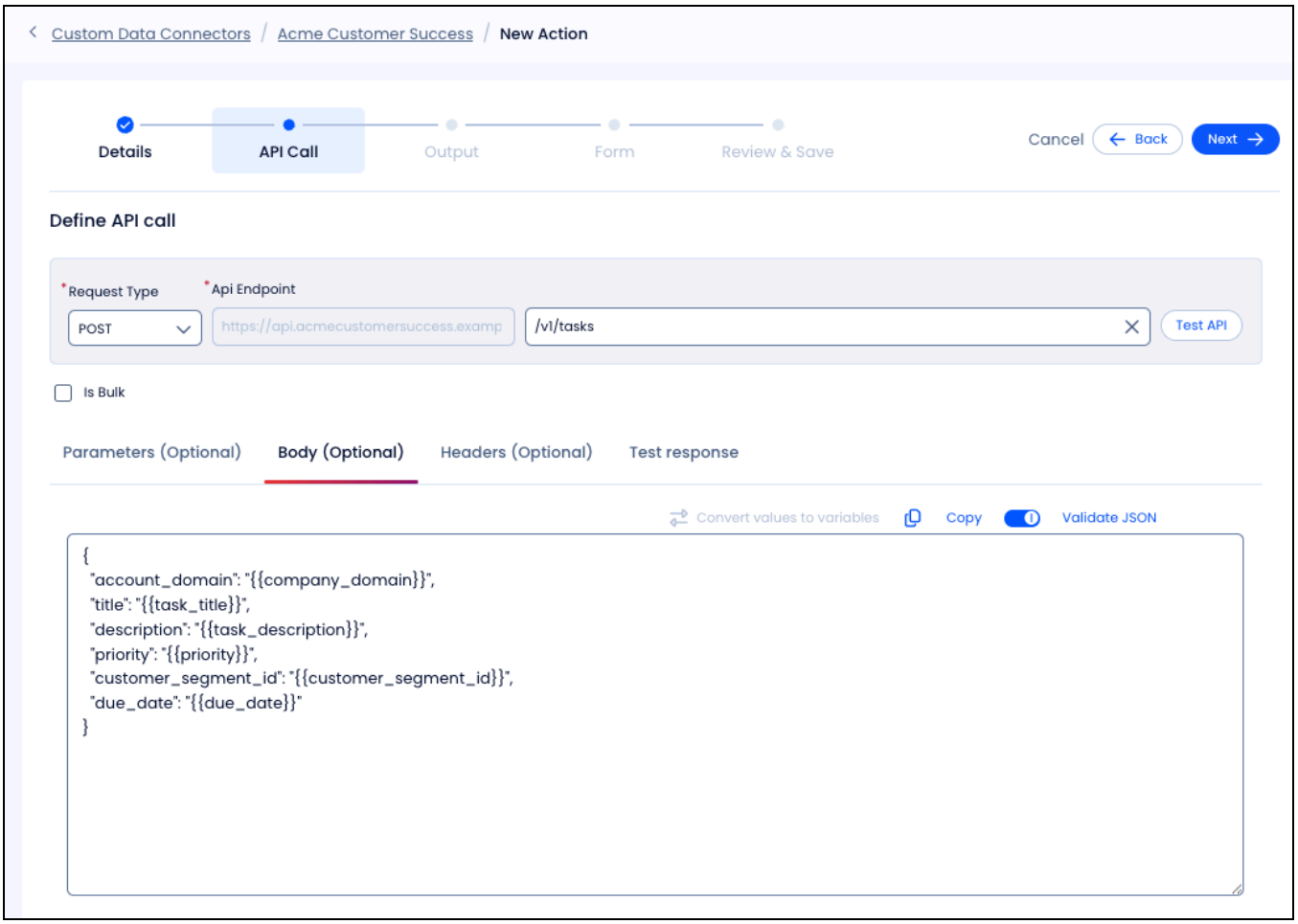
4. If required, select the **Headers (Optional)** tab and headers beyond the authentication configuration.



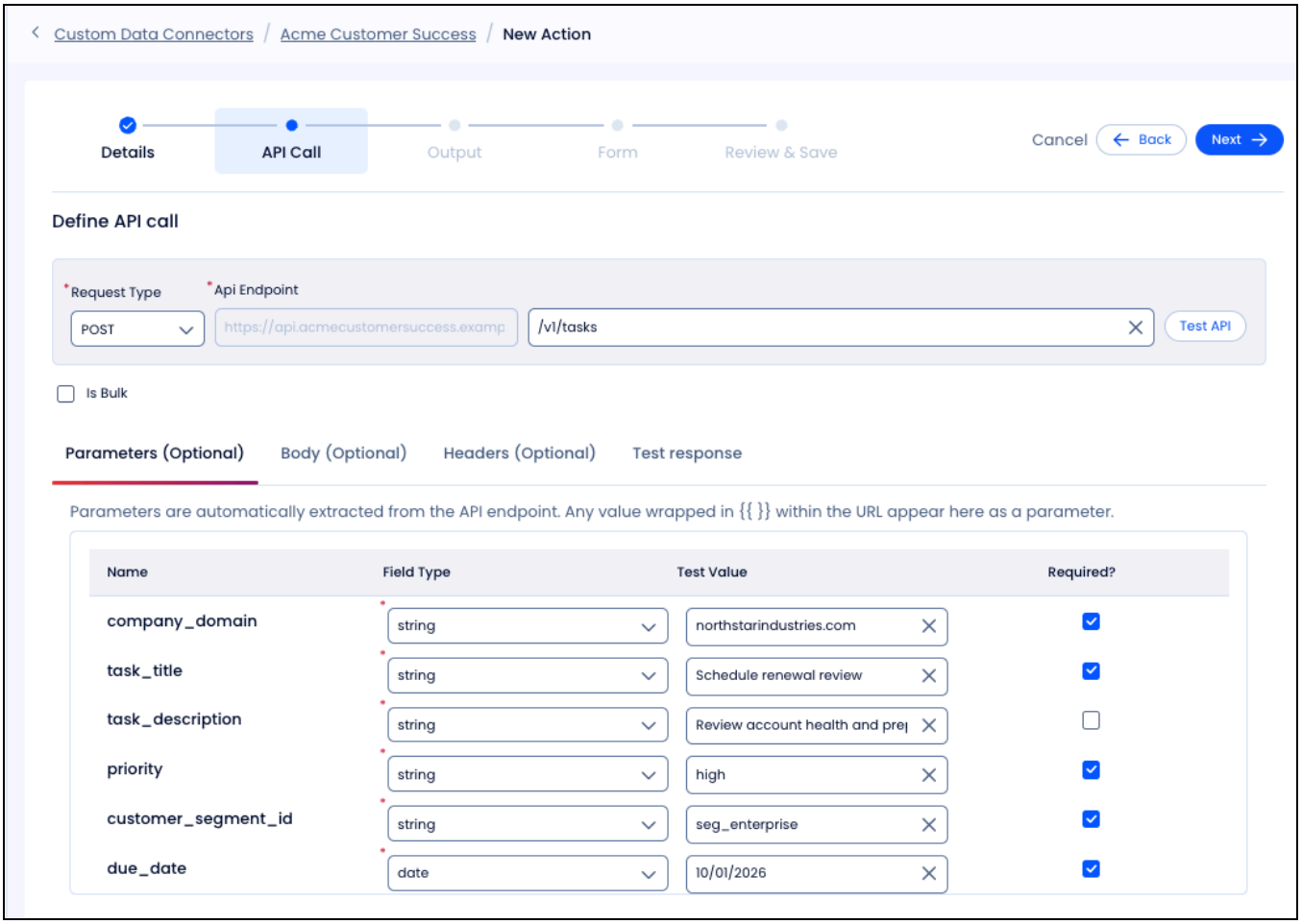
Step 3: Configure the request body

- For **POST, PUT, PATCH** requests, select the **Body (Optional)** tab and define the JSON request body.
- You can use `{{parameter}}` syntax to insert dynamic input values into the request body.
- The Action builder adds values referenced in the body to the **Parameters (Optional)** tab, where you can configure their field types and whether they are required.

Tip: Before creating a PUT or PATCH Action, confirm how the external API handles updates. Some APIs expect a complete record and may overwrite existing fields that are not included in the request. Test "update" Actions with non-production or test records whenever possible.



1. Configure any parameters on the **Parameters (Optional)** tab.
    - Select the **field type** (string, number, date, boolean, array)
    - Check whether the field is **required**



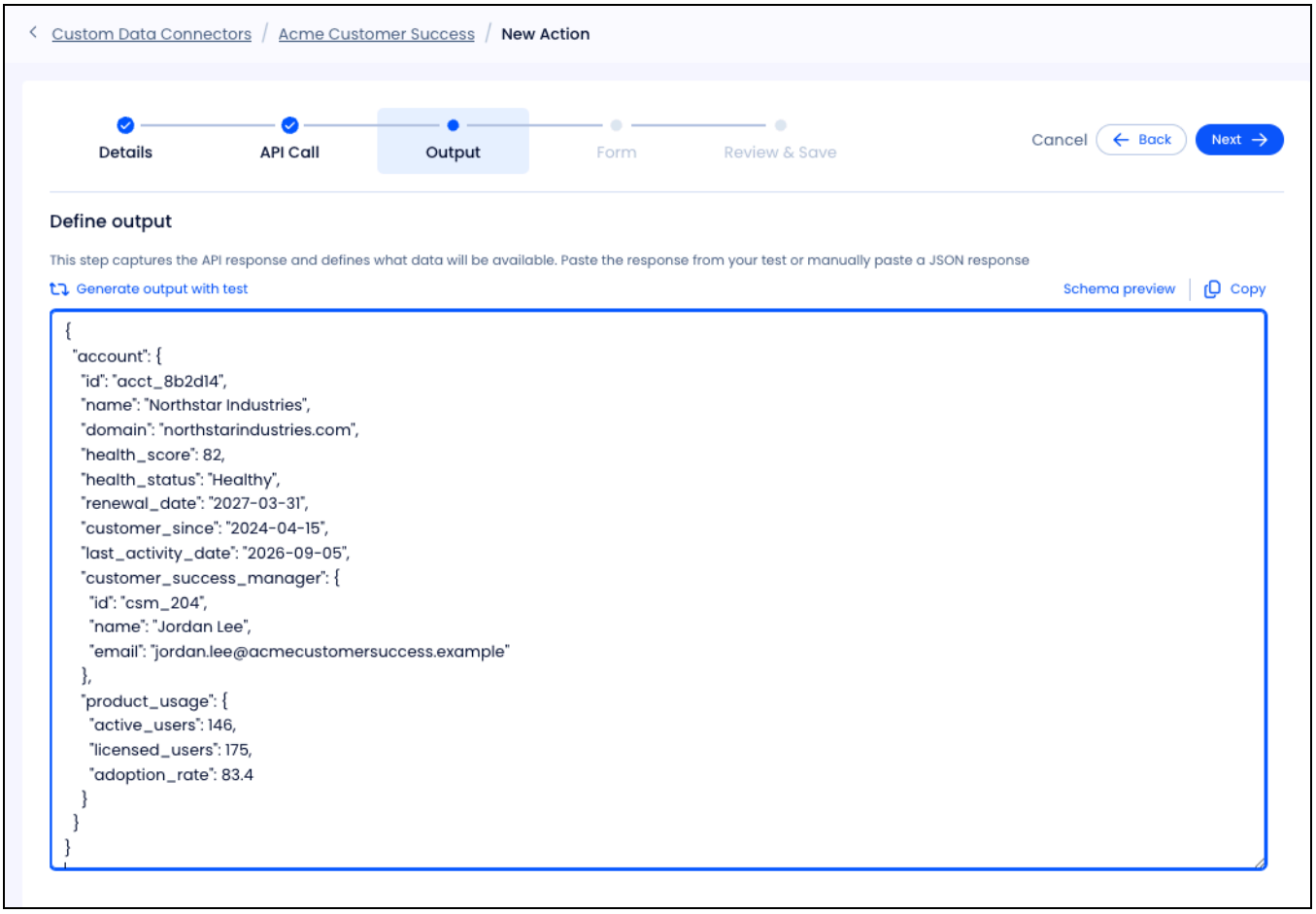
Step 4: Test the API call

Testing verifies that ZoomInfo can successfully call the endpoint and return the expected response.

1. Enter sample values for the parameters on the **Parameters (Optional)** tab
2. Select **Test API**. This will send an API request to the specified URL.
3. Review the result on the **Test response** tab. Confirm that:
    - Authentication succeeds.
    - Parameters are passed correctly.
    - The response includes the fields you expect.
    - The Action does not unintentionally modify external data.
4. Select **Next** when the API call returns the expected result.

Step 5: Define the Action Output

1. On the **Output** step, define which fields from the API response ZoomInfo makes available after the Action runs. You can:
    - Select **Generate output with test** to use the test response.
    - Paste in a sample JSON response from the external system.
    - Edit the response, for instance, by removing nodes or fields that users do not need.
    - Select **Schema Preview** to review the output fields ZoomInfo will create.



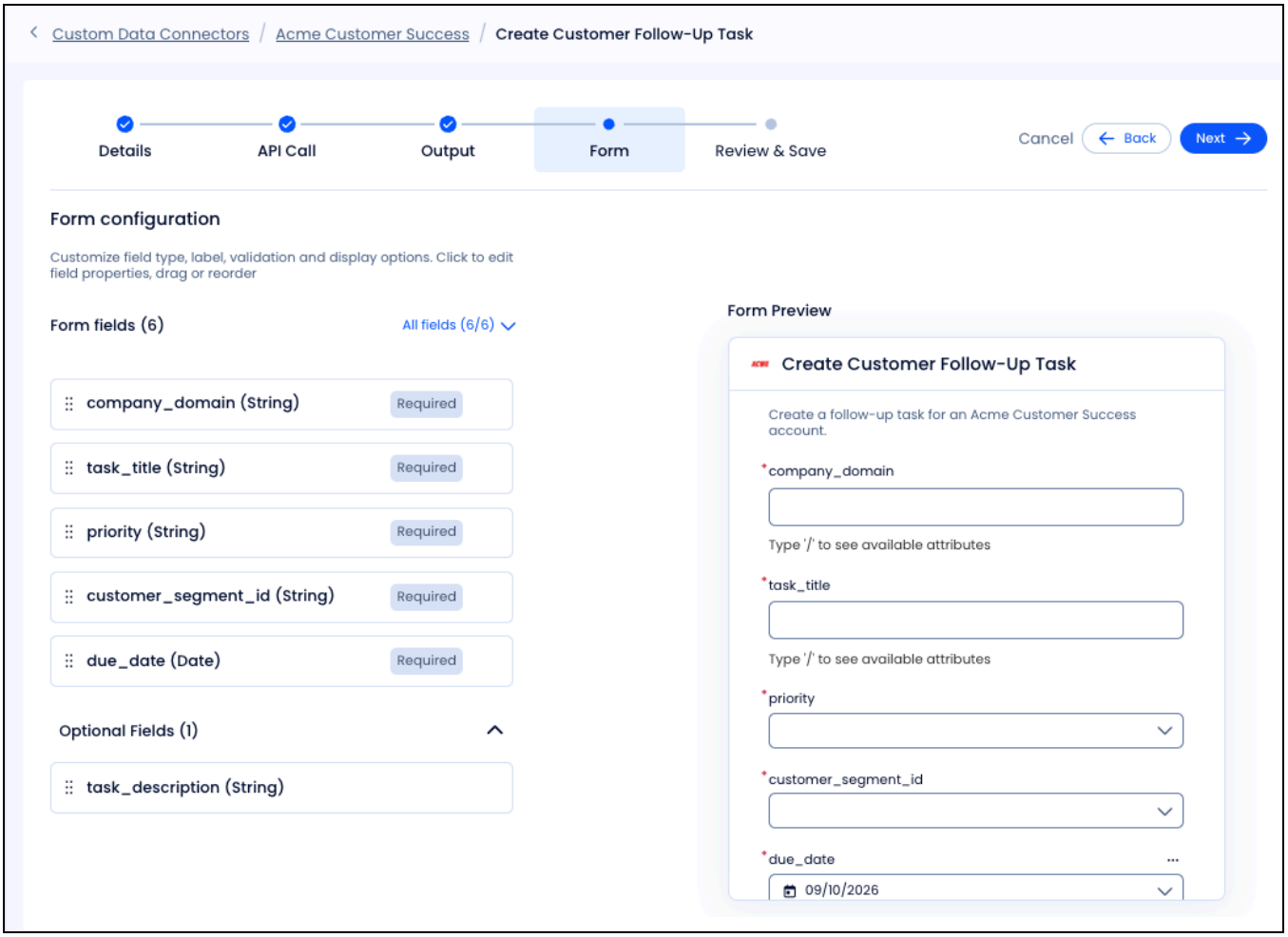
2. Select **Next** when the output schema is complete.

Tip: Keep the output focused on fields users need for enrichment or downstream Actions.

Step 6: Customize the Action Form

Use the form to control how users enter or map the parameters required by the external API. You can rename fields, change their display order, add instructional text, and configure dropdowns.

1. On the **Form** step, optionally customize the fields users see when they run the Action.



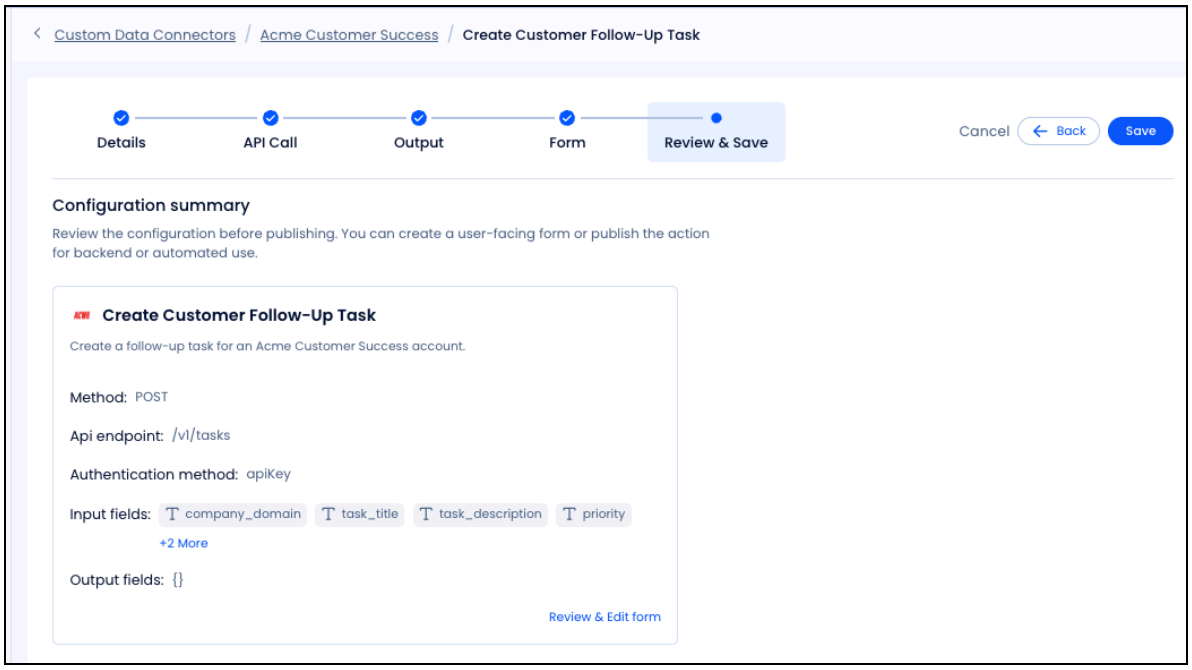
For detailed instructions, see **Customize the Action Form**.

2. Select **Next**.

Step 7: Review and save

On the **Review and Save** step,

- Review the Action details, API configuration, output schema, and Form Configuration.
- Select **Create** to make the Action available to ZoomInfo users.
- Repeat this process to add additional Actions to the connector.



Customize the Action Form

The **Form** step controls how Action inputs appear to ZoomInfo users. Customize the form when users need to enter a value, select an option, or map ZoomInfo data to the parameters required by the external API.

For example, a form can let a user:

- Enter an account ID, email address, or company domain.
- Map a field from an Audience to an Action parameter.
- Select a value from a predefined list.
- Select a value retrieved dynamically from the external system.

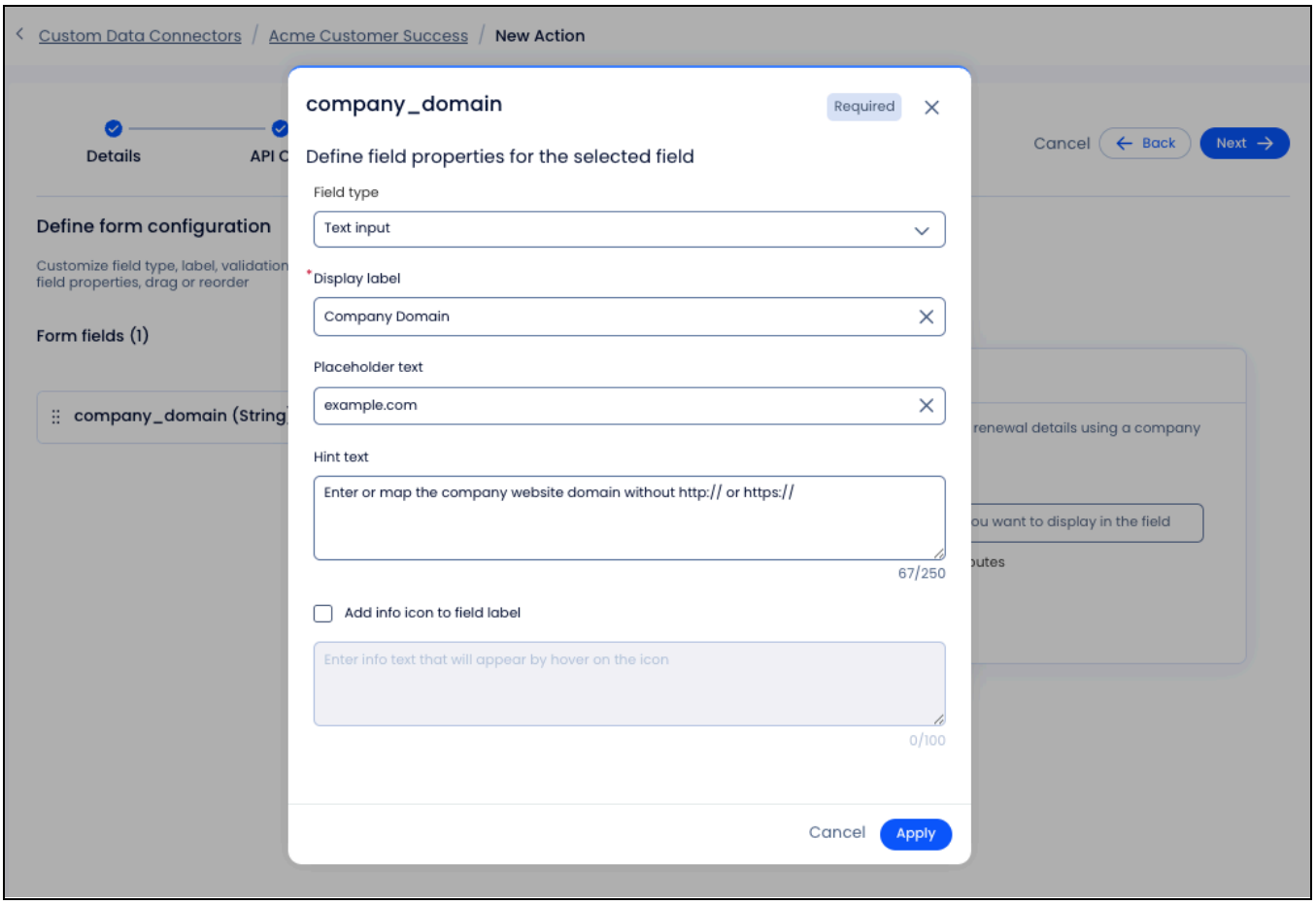
Form customization is optional. If you do not customize the form, ZoomInfo uses the Action parameters as configured during the API Call step.

Configure form fields

1. On the Action builder's **Form** step, use **Form Configuration** to review the available parameter fields.
2. Drag and drop fields to change the order in which they appear.
3. Select the edit icon next to a field to customize it.
4. Choose a **Field type**
    - **Text input:** For short values, such as an ID, name, email address, or domain.

- **Text area:** For longer values, such as notes, descriptions, or message content.
- **Dropdown:** For values users should select from a list.

5. Enter a **Display label**. This is the field name that users will see when they run the Action.
6. Optionally add:
    - **Placeholder text** to show an example or expected format.
    - **Hint text** to provide additional instructions.
    - **Info icon text** to provide contextual help.



7. Select **Apply**.
8. Use **Form Preview** to confirm that:
    - Fields appear in the intended order.
    - Labels and instructions are clear.
    - Required fields are correctly identified.

- Dropdowns show the expected options.
- Dynamic dropdowns return the expected list from the external system.



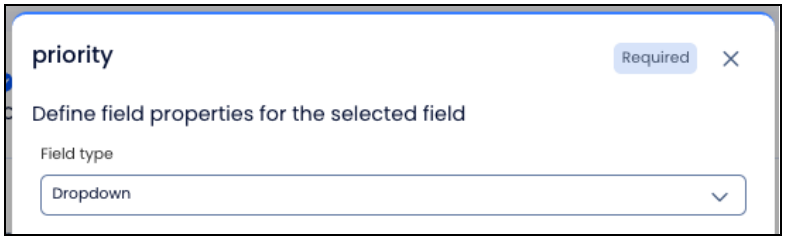
Form design best practices

1. Use labels that describe the information users should provide, rather than API-specific field names where possible.
2. Place fields in the order users naturally complete them.
3. Mark only necessary parameters as required.Use a dropdown when a controlled list of choices can reduce input errors.
4. Add placeholder or hint text when users need a particular format, such as a company domain or external record ID.
5. Use **Form Preview** to confirm the form is clear and usable before saving it.

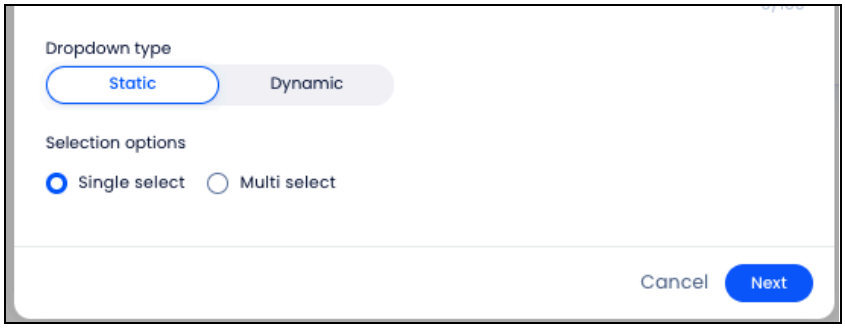
Configure dropdown field

Use a dropdown when users should choose from a list of allowed values instead of typing a value manually.

1. Edit the form field and select **Dropdown** as the field type.



2. Select the dropdown type:
    - **Static**: You define the available options.
    - **Dynamic**: ZoomInfo retrieves available options from an Action on the same connector.
3. Select whether users can choose:
    - **Single select**: Users select one value.
    - **Multi select**: Users select one or more values.
4. Select **Next** and configure the dropdown options.



Tip: Use a static dropdown for stable, predefined values. Use a dynamic dropdown when the available values are maintained in the external system and may change over time.

Configure a static dropdown

A static dropdown contains options that you define in the Action Form. You can either upload a CSV file or manually add options.

Upload options from a CSV file

Upload a CSV file with two columns and no header row:

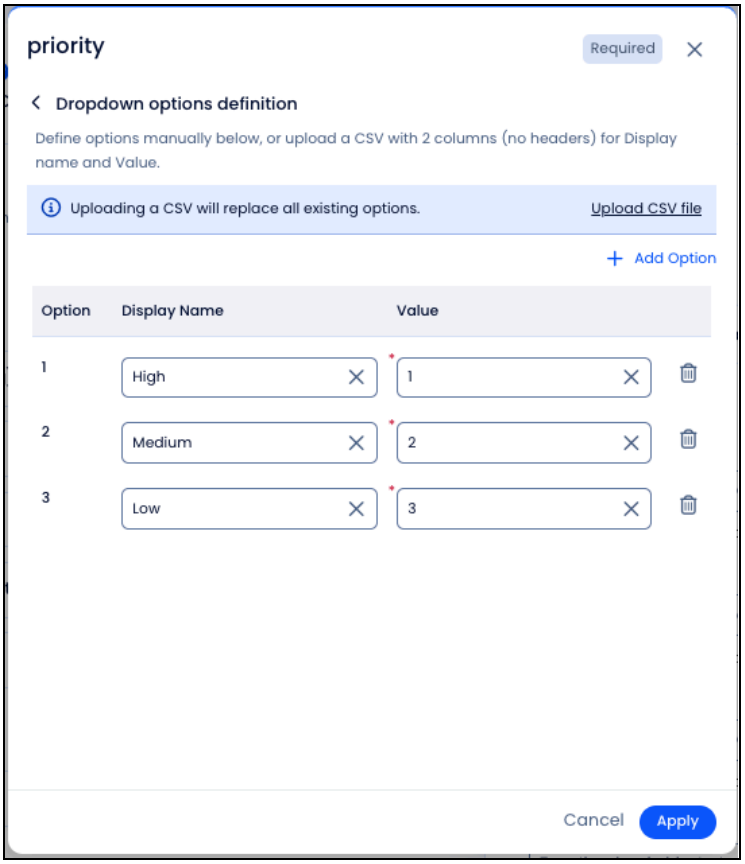
1. **Display name**: The label users see in the dropdown.
2. **Value**: The value ZoomInfo sends to the external API.

	A	B
1	High	1
2	Medium	2
3	Low	3

Add options manually

1. Enter a **Value**. This is the value ZoomInfo sends to the external API.
2. Optionally enter a **Display Name**. This is the label users see in the dropdown.
3. Select **Add Option** to add another item.

4. Use the delete control to remove an item.
5. Select **Apply** to save the dropdown configuration.



If you do not enter a display name, users see the value itself.

Configure a dynamic dropdown

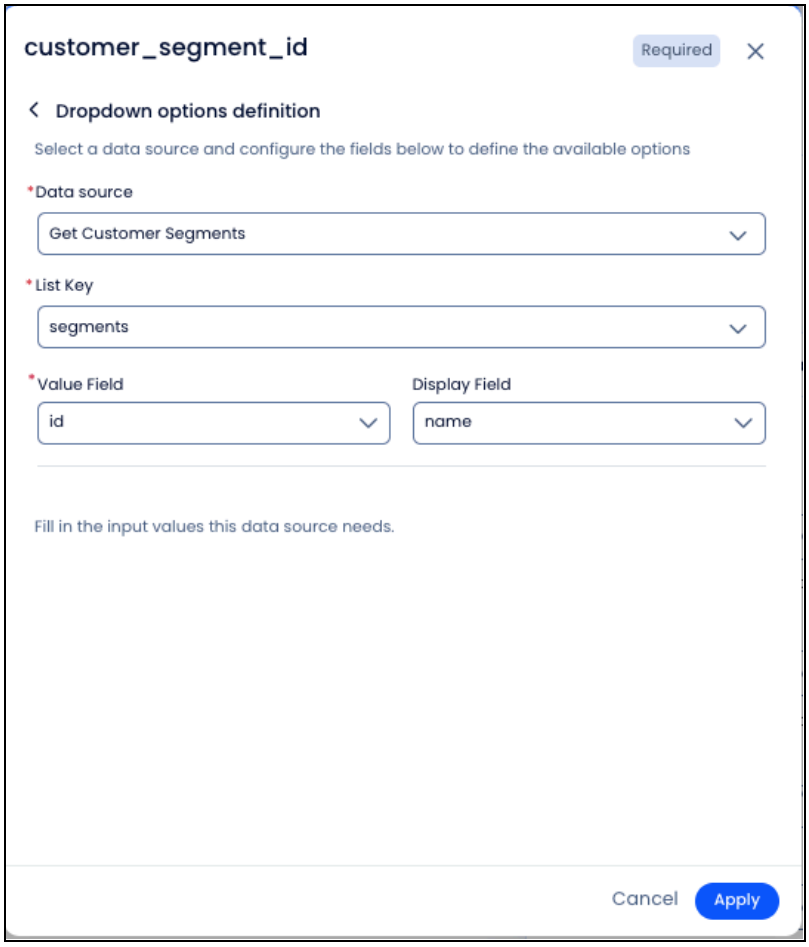
A dynamic dropdown retrieves its options from an external API through another Action on the same connector.

For example, a user might select a project, segment, owner, or region from values returned by the external system.

Before configuring a dynamic dropdown, create and test the Action that retrieves the list of available options.

1. Edit the form field and select **Dropdown**.
2. Select **Dynamic**.
3. Select a **Data Source** Action.
    - This is the Action that will generate the list. Only Actions that belong to the same connector are available.

4. Select the **List Key**.
    - This is the key in the response JSON which contains the list of dropdown items.
5. Select the **Value** key.
    - This required field is the value ZoomInfo sends to the external API after a user makes a selection.
6. Optionally select the **Display Name** keys.
    - This is the user-friendly label shown in the dropdown.



7. If the data-source Action requires parameters, provide values for them:
    - Enter a fixed value, or
    - Use the expression input to map a value dynamically from another form field.
8. Select **Apply**.
9. Use the **Form Preview** to confirm the Action is clear and usable before saving it.
10. Select **Next** to continue to **Review and Save**

11. Select **Save** to publish the Action and its Form Configuration.

Use a Connector Action in an Audience

After an administrator creates and saves a connector Action, ZoomInfo users can use the Action to enrich an Audience.

1. In an Audience, select **Actions**.
2. Select **View All Enrichments**.
3. Select **Connectors**, or search for the connector by name.
4. Select the connector, then select the Action you want to use.
5. Complete or map the Action form fields.
6. Run the Action to retrieve the configured output fields from the external system.

The available inputs and output fields depend on how the Action was configured. For example, users may map a company domain to an Action input and return external account attributes as enrichment fields.