Ecwid Events

SAP Cloud Platform Open Connectors currently supports events via **polling** or **webhooks** depending on the endpoint.

- **Polling** is a mechanism where SAP Cloud Platform Open Connectors executes the configured query every `n` minutes and captures the changed information.
- **Webhooks** are when the prider lets SAP Cloud Platform Open Connectors know what information has changed. Note that additional endpoint setup may be required prior to creating your Connector Instance.

> **ⓘ Note:** SAP Cloud Platform Open Connectors normalizes only the objectId, objectType, and eventType event data. Event data also contains raw data, which the provider returns. The raw data varies based on the endpoint.

If you would like to see more information on our Events framework, see Events Overview .
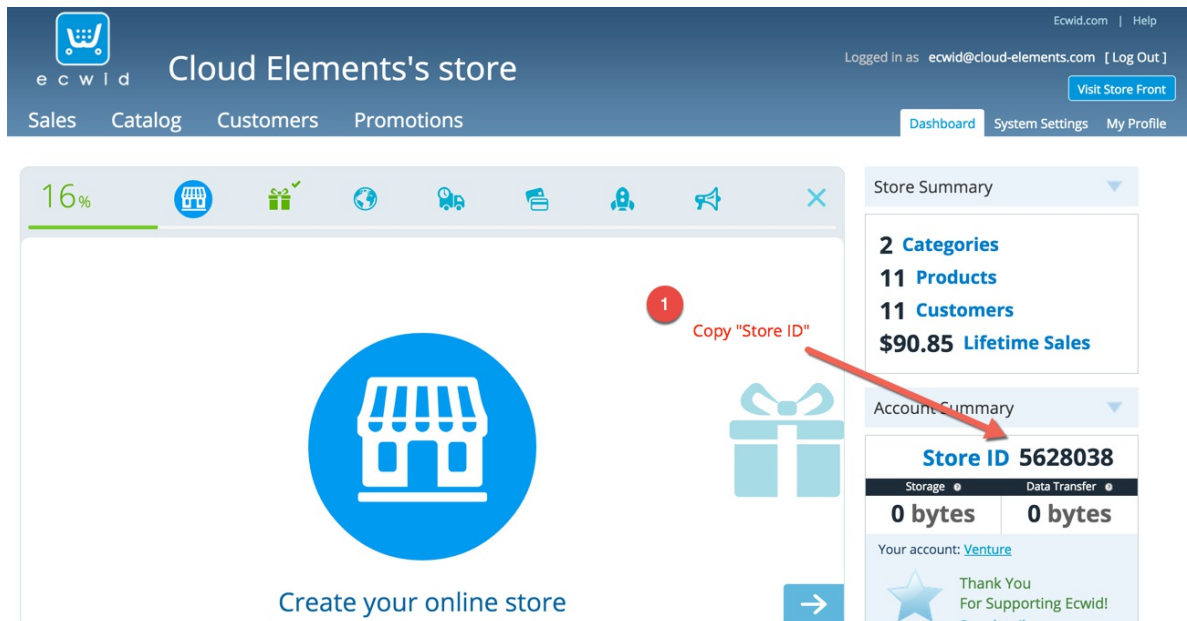
SAP Cloud Platform Open Connectors supports webhook events for Ecwid.

Webhooks

Follow these steps to set up a new Ecwid Application for API integration. Via a web browser go to: https://my.ecwid.com/cp/#register and sign up. It must be a paid account.

Once setup, please login.

1. Make note of your Store ID as it will be needed to provision a connector Instance.
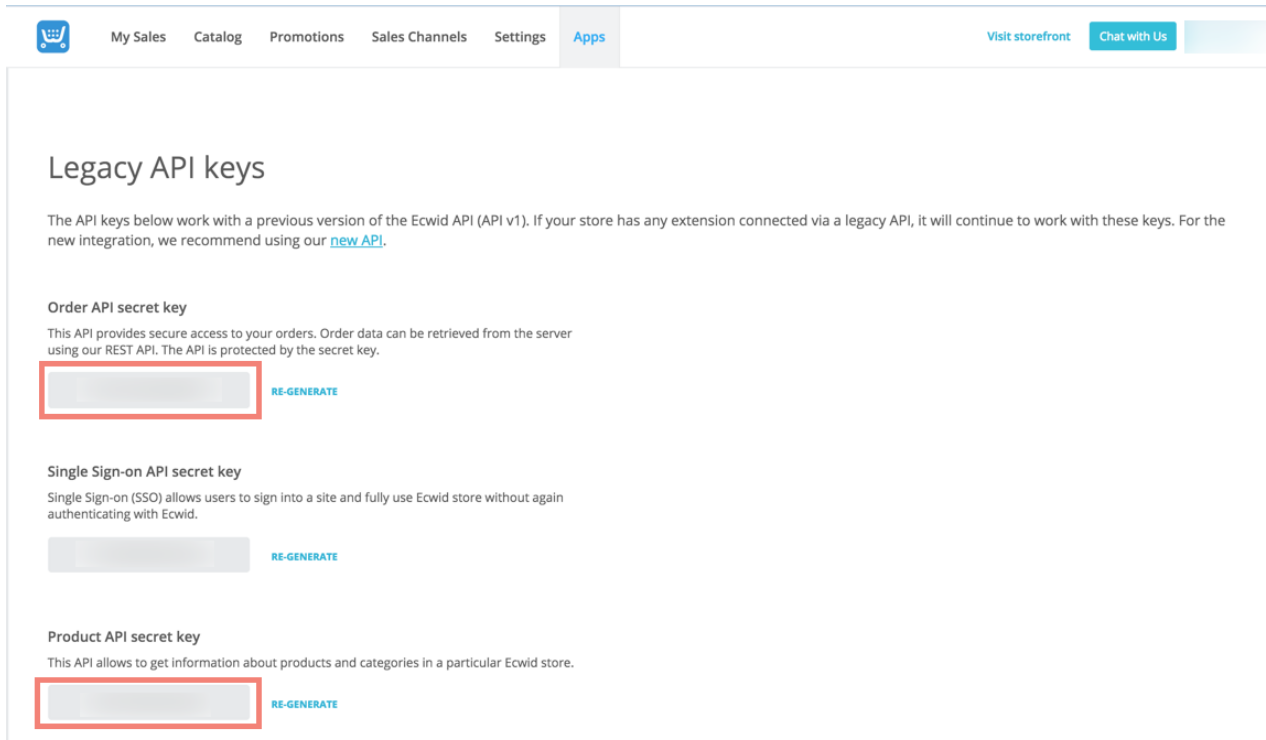
2. Next the Legacy API keys will need to be retrieved. In order to do retrieve them, you must be logged in to your store then navigate to a Legacy API Key URL

NOTE: Your store ID is required in the Legacy API Key URL - replace `INSERT_STORE_ID` in the URL with your actual store ID.

Legacy API Key URL (Don't forget to insert your store ID):

```
https://my.ecwid.com/store/INSERT_STORE_ID#legacy_api
```

Copy the Order and Product API secrets

If you wish to add event functionality then please enter the following URL in the ION Cannon endpoint URL:

```
https://api.openconnectors.us2.ext.hana.ondemand.com/elements/api-
v2/events/ecwid
```

ION Cannon endpoint URL

| https://api.cloud-elements.com/elements/api-v2/events/ecwid | Save |

TEST ENDPOINT...

Ecwid Orders are currently supported within the Events Framework.

In order to enable webhooks, add these two extra configurations to your instance JSON:

```
"event.notification.enabled": "true",
"event.notification.callback.url": ""
```

instance JSON with webhook events enabled:

```
{
  "element": {
    "key": "ecwid"
  },
  "configuration": {
    "ecwid.order.key": "",
    "ecwid.product.key": "",
    "ecwid.store.id": "",
    "event.notification.enabled": "true",
    "event.notification.callback.url": ""
  },
  "tags": [
    ""
  ],
  "name": ""
}
```