

Installing the Stripe app for Adobe Commerce

Overview

This guide walks you through installing the Stripe app for **Adobe Commerce** via the Adobe Exchange marketplace. The app integrates Stripe's payment processing with Adobe Commerce to support modern, secure, out-of-process payment processing; without modifying your core Commerce application. Instead of installing modules directly into the codebase (in-process extensibility), it supports out-of-process extensibility using Adobe App Builder. These apps run securely and independently in Adobe I/O Runtime and integrate with your Adobe Commerce Storefront via the GraphQL APIs and Adobe I/O events.

Supported platforms

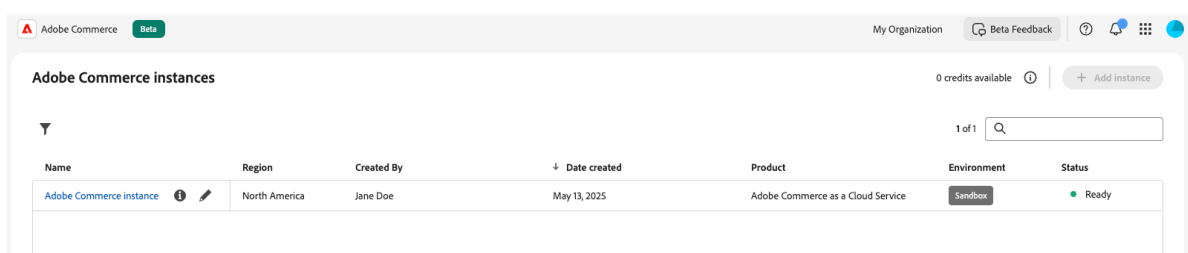
The Stripe app currently supports Adobe Commerce as a Cloud Service (SaaS). Support for Adobe Commerce on cloud infrastructure (PaaS) and Adobe Commerce on-premises is coming very soon.

Before you begin

Set up Adobe Commerce as a Cloud Service

Before you begin, you will first need to have an instance of Adobe Commerce as a Cloud Service running. You can create one by:

1. Navigating to Experience Cloud Console
2. Under **Quick Access**, click **Commerce**
3. At the top right hand side of the page, click **Add instance**



If you are testing the app on a sandbox / development environment, you can import available sample data from Adobe to your Commerce instance.

Set up your Stripe account

You can register for a Stripe account via Stripe's website. After registering, you can access your Stripe API keys via:

- Test Mode: <https://dashboard.stripe.com/test/apikeys>
- Live Mode: <https://dashboard.stripe.com/apikeys>

Set up App Builder

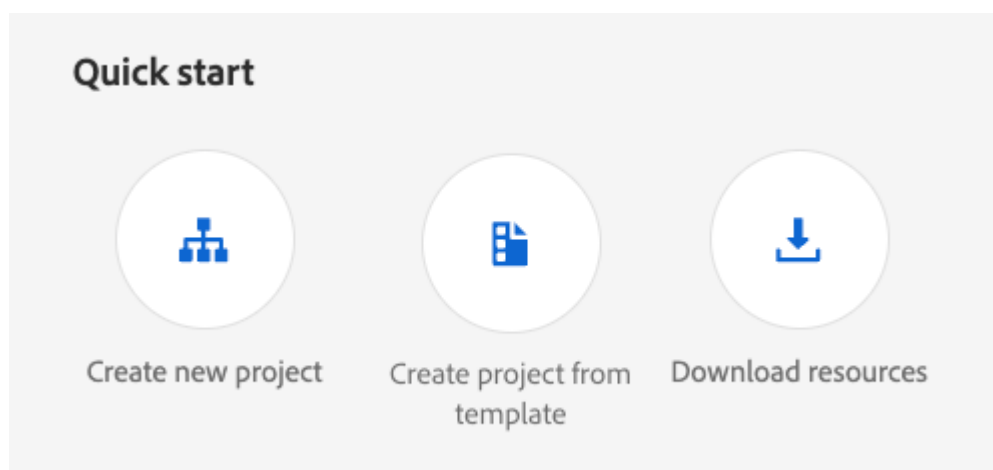
Obtain a license

You will need a license for App Builder in order to deploy the app. App Builder is necessary so that you can configure IMS authentication credentials, configure Commerce I/O Events, persist data to Adobe I/O Lib Files, or generate encryption keys for your environment file.

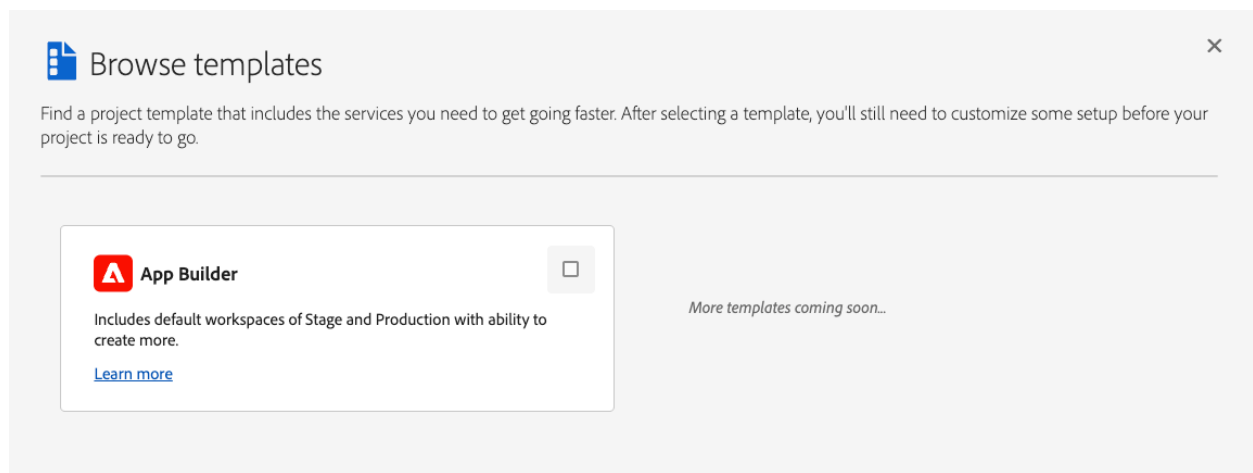
You can sign up to a trial of App Builder via <https://developer.adobe.com/app-builder/trial/>.

Create the project

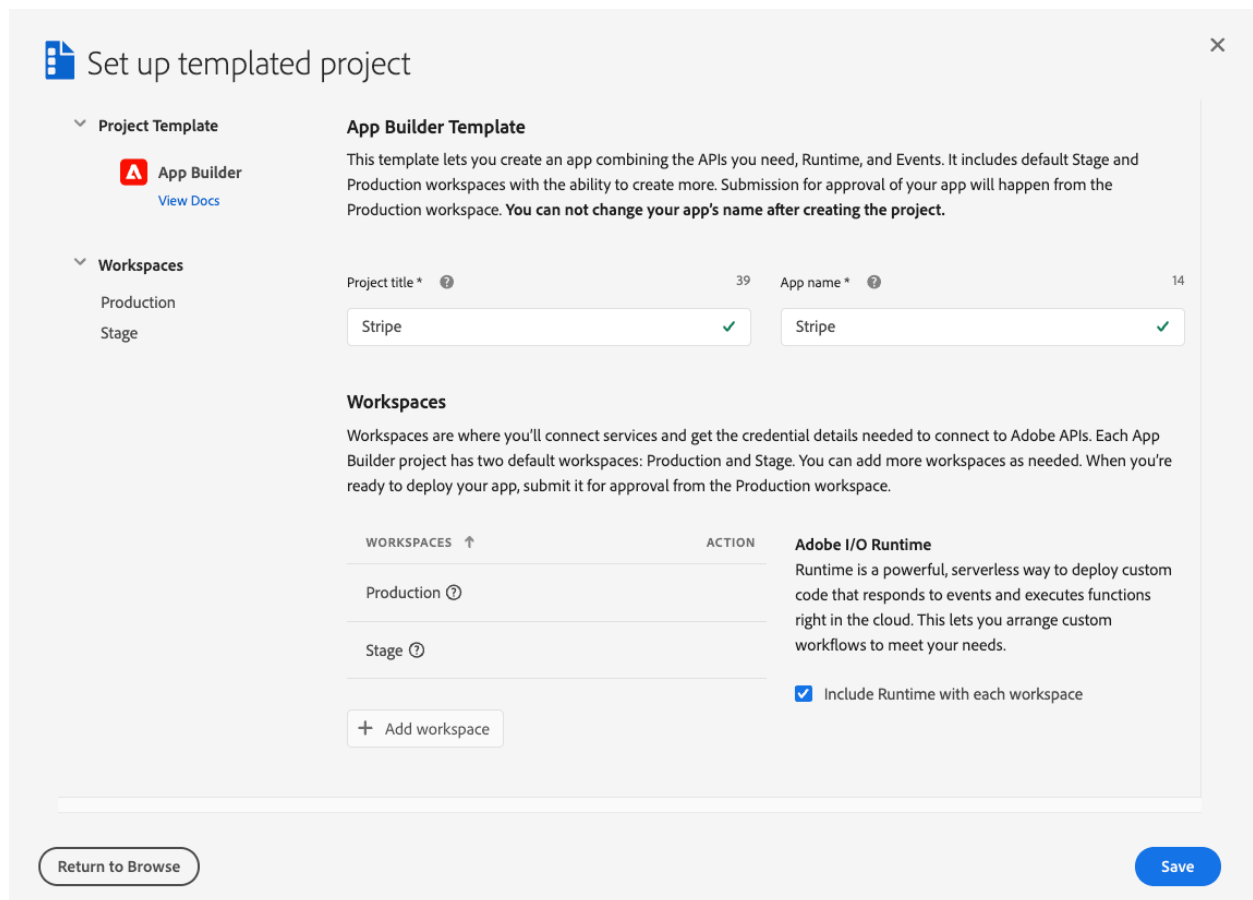
Once you have an App Builder license, navigate to Adobe Developer Console. You should see there a **Create project from template** button, indicating that your App Builder license is active:



Click on the **Create project from template** button, and select App Builder as a template:

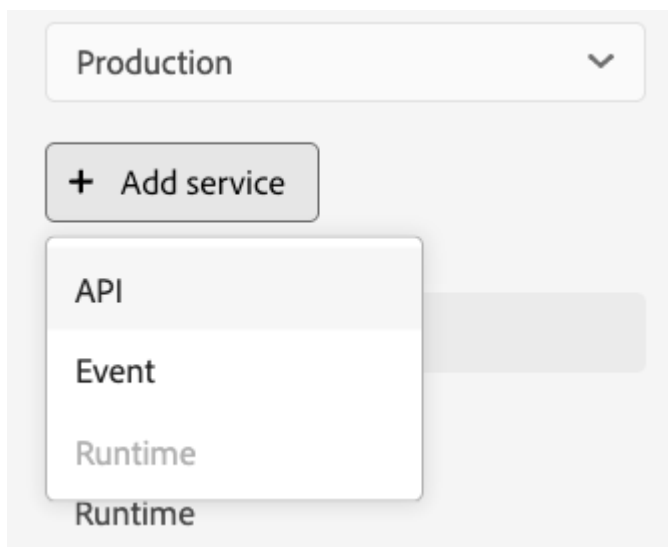


Fill the project details and click save:

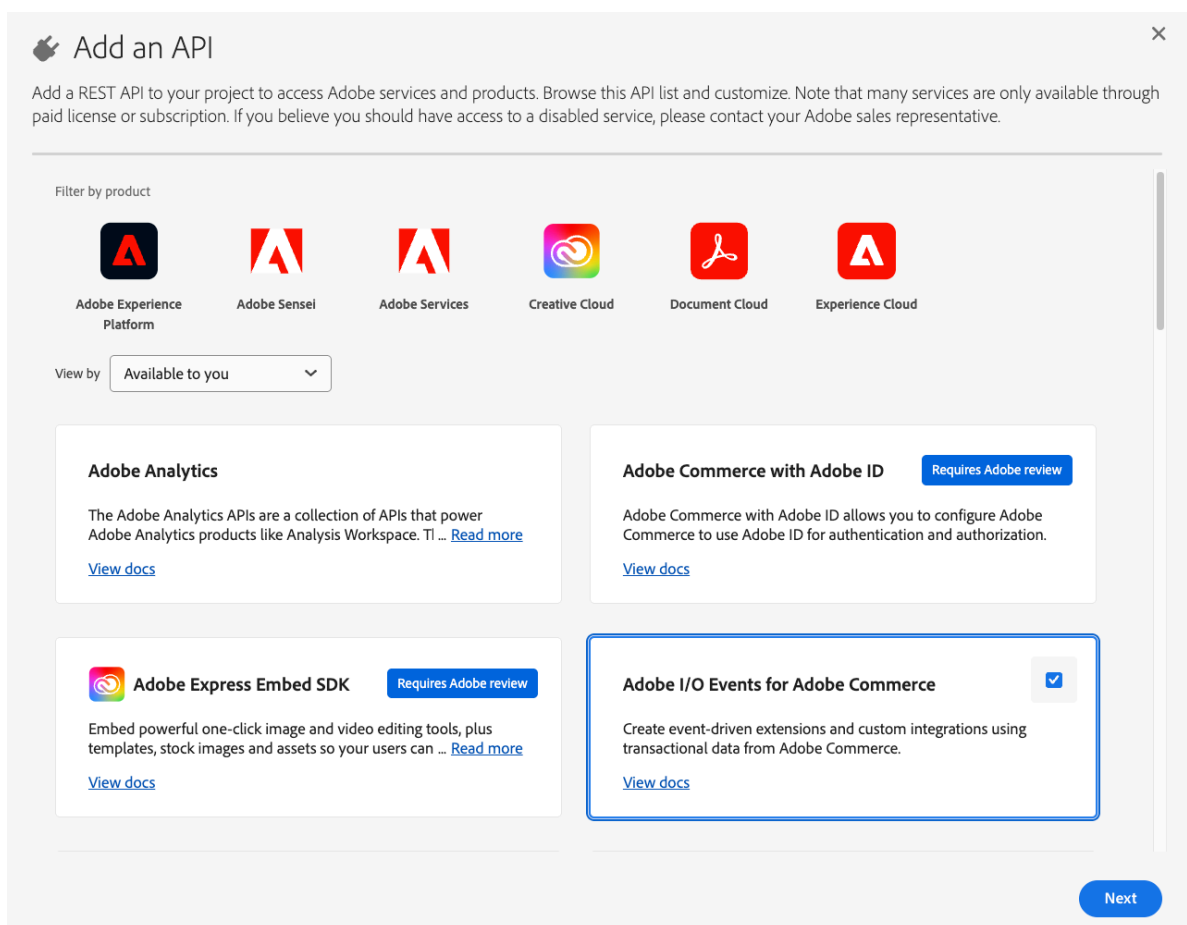


Add necessary APIs to the project

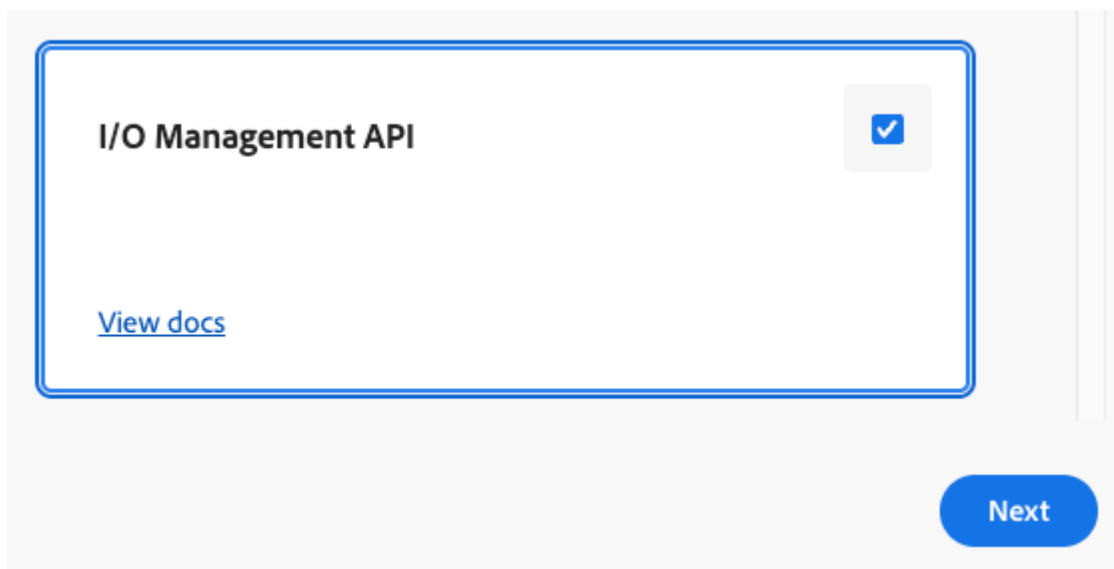
1. Select an environment, such as **Stage** or **Production**, click the **Add service** button, and then select **API**.



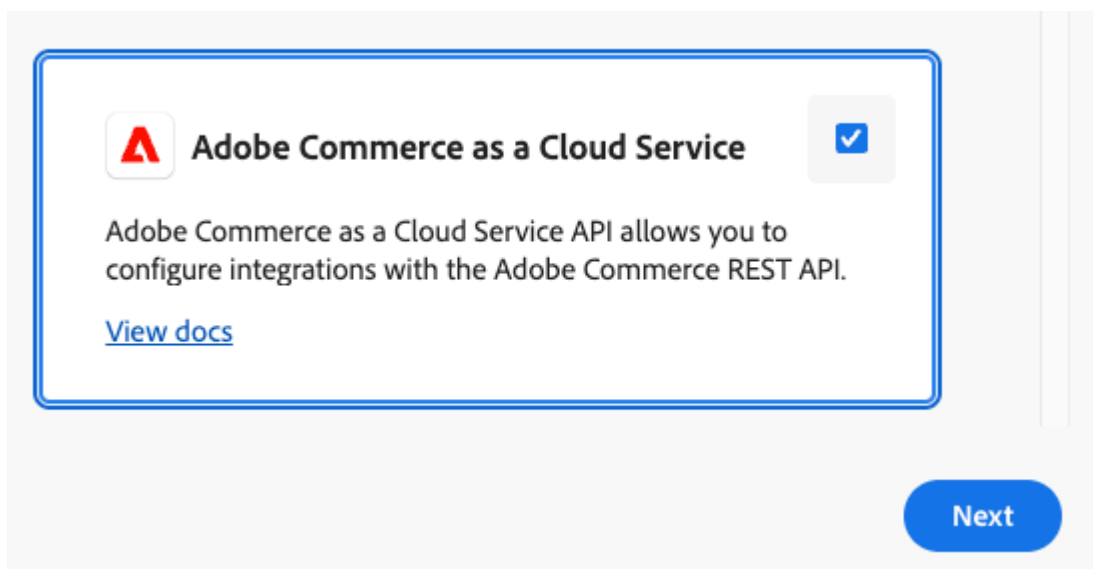
2. Select **Adobe I/O Events for Adobe Commerce** and add it to the environment.



3. Repeat the same steps, and add the **I/O Management API** to the environment.



4. Repeat the same steps, and add the **Adobe Commerce as a Cloud Service** to the environment.



Set up your local environment

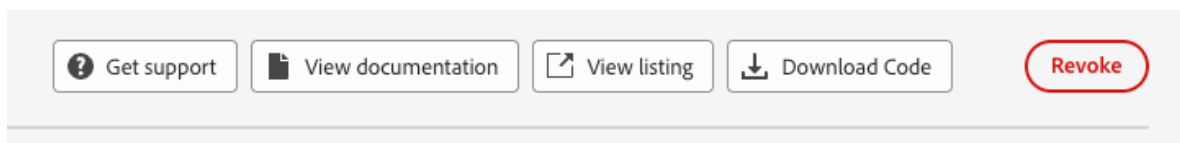
Before you can deploy the app to App Builder, you will additionally need to install **Node.js** and the **Adobe I/O CLI** package. Please follow the official instructions to set up your local environment.

Install the App via Adobe Exchange Marketplace

To install the Stripe app, follow the steps below:

1. Navigate to the app's listing on the Adobe Exchange Marketplace
2. Click on the app listing to view it.
3. At the top right hand side, click the **Free** button and follow the instructions to install the app into your organization's environment.

4. The app will appear under Adobe Exchange App Management. If you are an account admin, click the **Review** button on the app page in order to approve it.
5. At the app's page, you will see a **Download Code** button at the top bar. Click it to download the app's code locally. You will need the code to set up your OAuth Credentials.



6. Extract the downloaded code to a local directory, and use the CLI to run the following commands:

```
# Navigate to the extracted project directory
cd path/to/extracted/stripe-app

# Login and configure Adobe CLI (run from the project root directory)
aio login -f
aio console org select
aio console project select
aio console workspace select
aio app use --merge
```

7. Confirm that the `.env` and `.aio` files were added to the project directory, and run the following command:

```
# Run from the Stripe app project root directory
npm run sync-oauth-credentials
```

This will generate entries in the `.env` file:

```
OAUTH_CLIENT_ID=
OAUTH_CLIENT_SECRETS=[""]
OAUTH_TECHNICAL_ACCOUNT_ID=
OAUTH_TECHNICAL_ACCOUNT_EMAIL=
OAUTH_SCOPES=[""]
OAUTH_IMS_ORG_ID=
```

The above values will be read by the deployed app's runtime actions, in order to create access tokens for the Adobe Commerce Admin API.

8. Generate and add encryption keys for sensitive credentials by running the following command:

```
# Run from the Stripe app project root directory
npm run generate-encryption-key
```

The command will output two environment variables which you need to add to your `.env` file:

```
ENCRYPTION_KEY=
ENCRYPTION_IV=
```

9. Install app dependencies:

```
# Run from the Stripe app project root directory
npm install
```

10. Deploy the updated app:

```
# Run from the Stripe app project root directory
aio app deploy
```

Stripe Configuration

Before you can access the Stripe configuration page, you will need to register the Stripe app in your Adobe Commerce admin area. You can do that by following the steps below:

1. Log into your Adobe Commerce admin area.
2. Navigate to **Stores > Settings > Configuration > Adobe Services > Admin UI SDK**

Configuration

Scope: Default Config

Save Config

GENERAL

SECURITY

CATALOG

CUSTOMERS

SALES

ADOBE SERVICES

Admin I/O Events

Admin UI SDK

Webhooks

AEM Assets Integration

SERVICES

General configuration

Enable Admin UI SDK [global] Yes

Enable the AdobeAdminIm module to use the Admin UI SDK.

[global] **Configure extensions**

Click to select extensions and the preferred workspace.

[global] **Refresh registrations**

Refresh registrations on schedule [global] Daily ☐ Use system value

Database logging configuration

Enable logs [global] No ☒ Use system value

Testing

Enable local testing [global] No

Copyright © 2025 Adobe. All rights reserved. Privacy Policy | Report an Issue

3. Enable the **Admin UI SDK** and set the **Refresh registrations on schedule** to daily.

4. Click **Configure extensions**.

Configuration

Scope: Default Config

Save Config

GENERAL

SECURITY

CATALOG

CUSTOMERS

SALES

ADOBE SERVICES

Admin I/O Events

Admin UI SDK

Webhooks

AEM Assets Integration

SERVICES

General configuration

Enable Admin UI SDK [global] Yes

Enable the AdobeAdminIm module to use the Admin UI SDK.

[global] **Configure extensions**

Click to select extensions and the preferred workspace.

[global] **Refresh registrations**

Refresh registrations on schedule [global] Daily ☐ Use system value

Database logging configuration

Enable logs [global] No ☒ Use system value

Testing

Enable local testing [global] No

Eligible Extensions

Choose the workspace and extensions you want to add to your Adobe Commerce Admin panel.

Workspace: Production **Apply**

Extension title Extension name Extension status **Apply Filters** **Clear Filters**

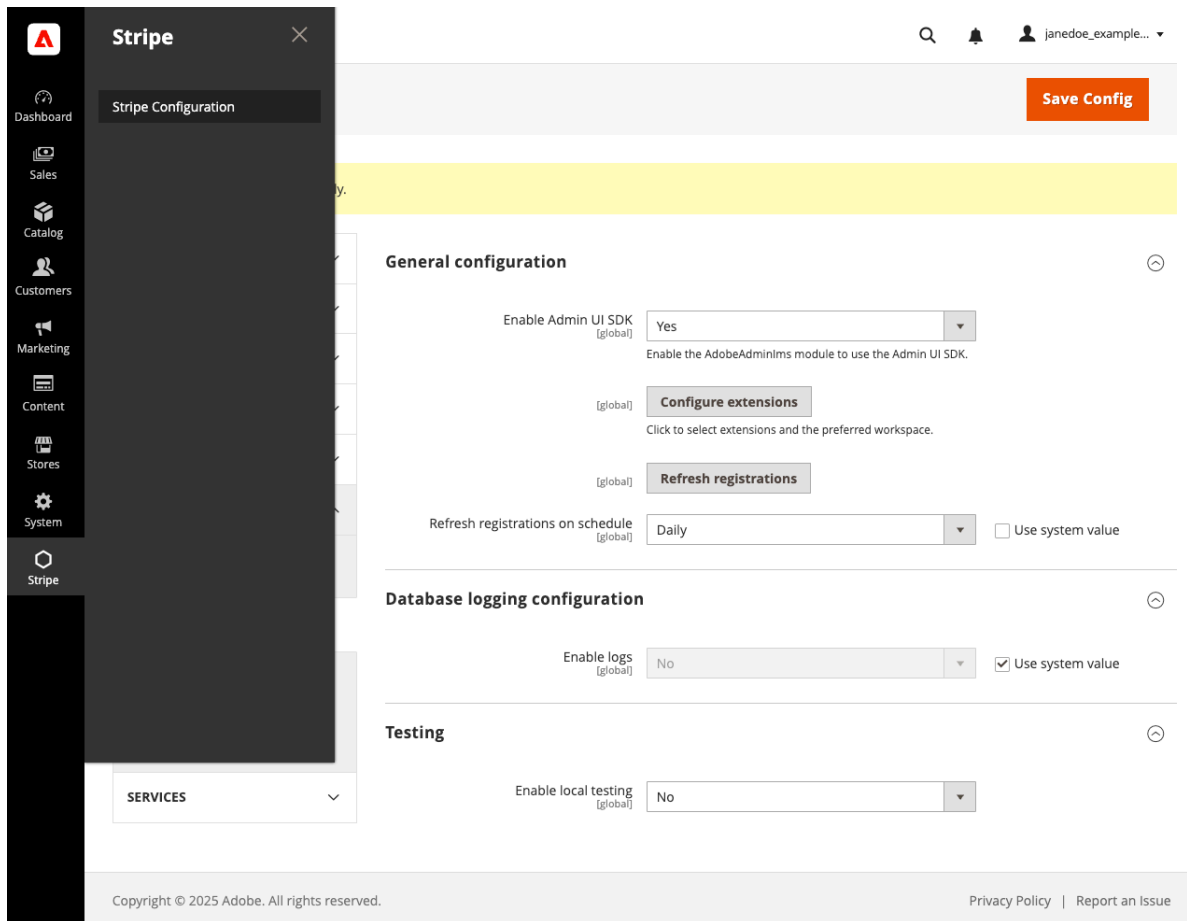
<input type="checkbox"/>	Extension title	Extension name	Extension status	Reference URL
<input checked="" type="checkbox"/>	650001-stripe	650001-stripe-production	Draft	https://650001-stripe-production.adobeio-static.net/index.html

Save **Close**

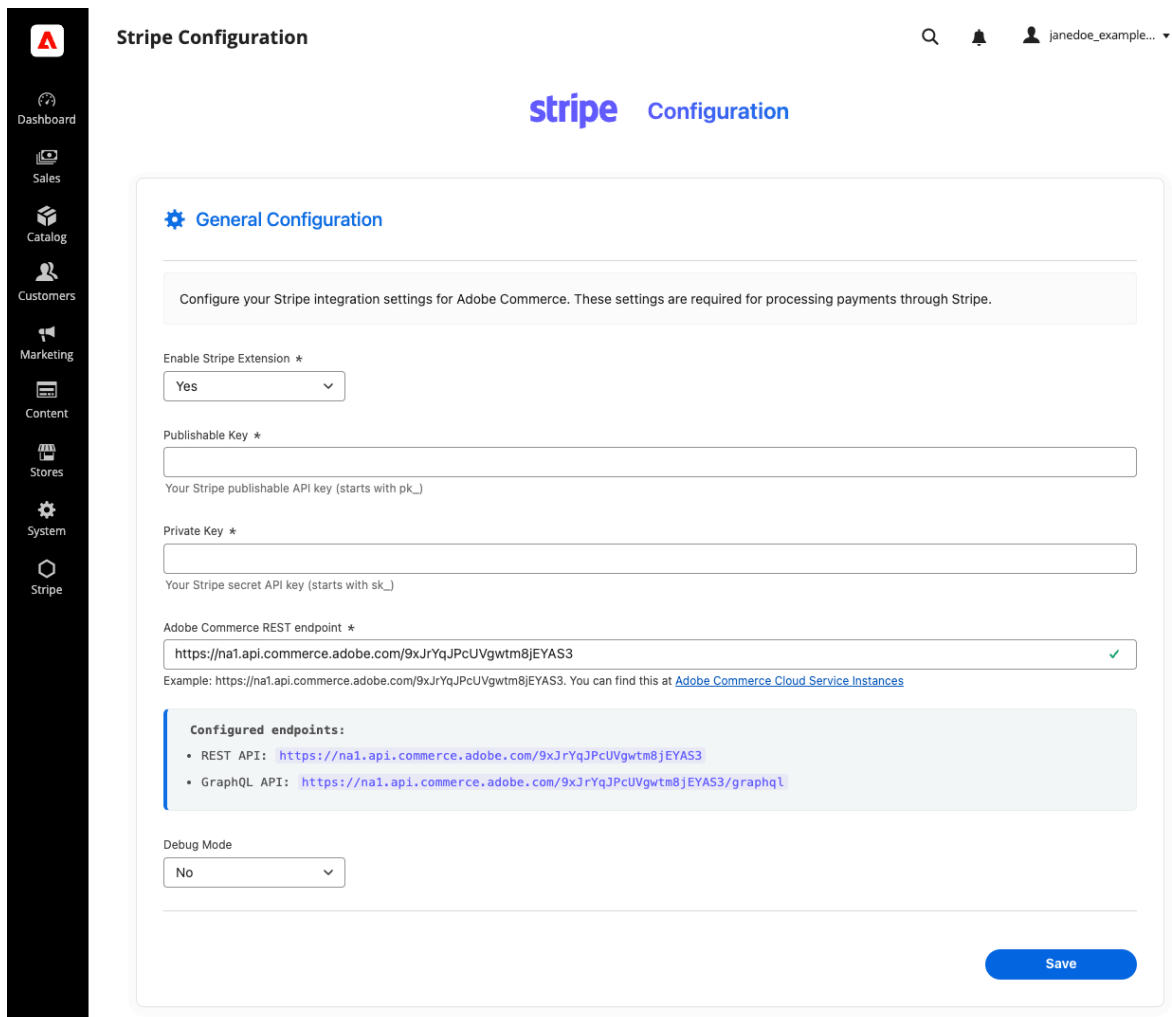
Copyright © 2025 Adobe. All rights reserved. Privacy Policy | Report an Issue

5. Select the Stripe app and click **Save**.

6. A Stripe menu will appear on the sidebar. Click **Stripe Configuration**.



7. The main configuration screen will appear.



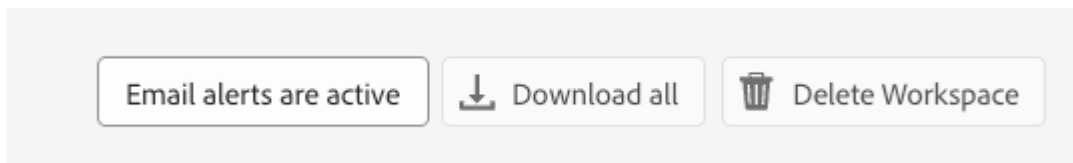
The following settings are available for configuration

- **Enable:** Will enable or disable the payment method at your storefront.
- **Publishable and Private Key:** These are the Stripe account API keys which will be used to collect payments. You can find these at <https://dashboard.stripe.com/apikeys>
- **Adobe Commerce URL:** This is the REST API endpoint of your Adobe Commerce as a Cloud Service instance. You can get this by navigating to your Commerce instances, and clicking the information icon (i) next to the Commerce instance you want to connect to. A modal will appear listing the **REST endpoint**.
- **Debug:** If you will be building your EDS storefront, Debug mode may return additional error details on various API calls to the app's runtime.

Events Configuration

With the app configured, the next step is to subscribe to events emitted by your Adobe Commerce as a Cloud Service instance. This will allow the app to update payments in your Stripe dashboard with order details, and automatically invoice orders that have been paid.

1. At the top right of the environment page, click **Download all**.



The downloaded file includes metadata in json format about the app. Open the file in a text editor and copy its contents.

2. Navigate to your Adobe Commerce admin under **Stores > Settings > Configuration > Adobe Services > Adobe I/O Events > General configuration**.

General configuration

Adobe I/O Authorization Type <small>[global]</small>	OAuth (Recommended) ▼ <small>The selected type must match the server-to-server authentication type defined in your App Builder project.</small>
Adobe I/O Workspace Configuration <small>[global]</small>
Adobe I/O Event Provider Instance ID <small>[global]</small>	 <small>This field identifies the Adobe I/O Events Provider linked to your Adobe Commerce Instance.</small>
Event Provider <small>[global]</small>	Create Event Provider
Adobe I/O Event Provider ID <small>[global]</small>	
Adobe I/O Event Provider Metadata <small>[global]</small>	dx_commerce_events <input checked="" type="checkbox"/> Use system value
Synchronize Events <small>[global]</small>	Execute Synchronization <small>Complete all fields in the "General configuration" section before synchronizing events.</small>

3. In the **Adobe I/O Workspace Configuration**, paste the contents of the metadata file that you just downloaded.
4. In the **Adobe I/O Event Provider Instance ID**, enter a unique identifier for your Commerce instance. You will connect to this provider ID when subscribing to events.
5. Click **Save Config** to save the configuration.
6. Click **Create Event Provider** to create the event provider. You will be asked to specify a label and description for it. Enter any values you like and click **Submit**.
7. Under the **Commerce events** section, set **Enabled** to **Yes** and save the configuration.

Commerce events

Enabled

global

Yes

☐ Use system value

Merchant ID

global

TeamMercury

Environment ID

global

DeveloperSandbox

Maximum retries to send events

global

7

☒ Use system value

Event retention time (in days)

global

3

☒ Use system value

Test Connection

global

Send Test Event

8. You can optionally test the connection by clicking **Send Test Event**.
9. Navigate to System > Events > Events Subscriptions. Click **Add new subscription**, and add the **observer.checkout_oope.checkout_submit_all_after** event.

Events Subscriptions

jane_doe

Add New Subscription

Columns

1 records found

20

▼

per page

<

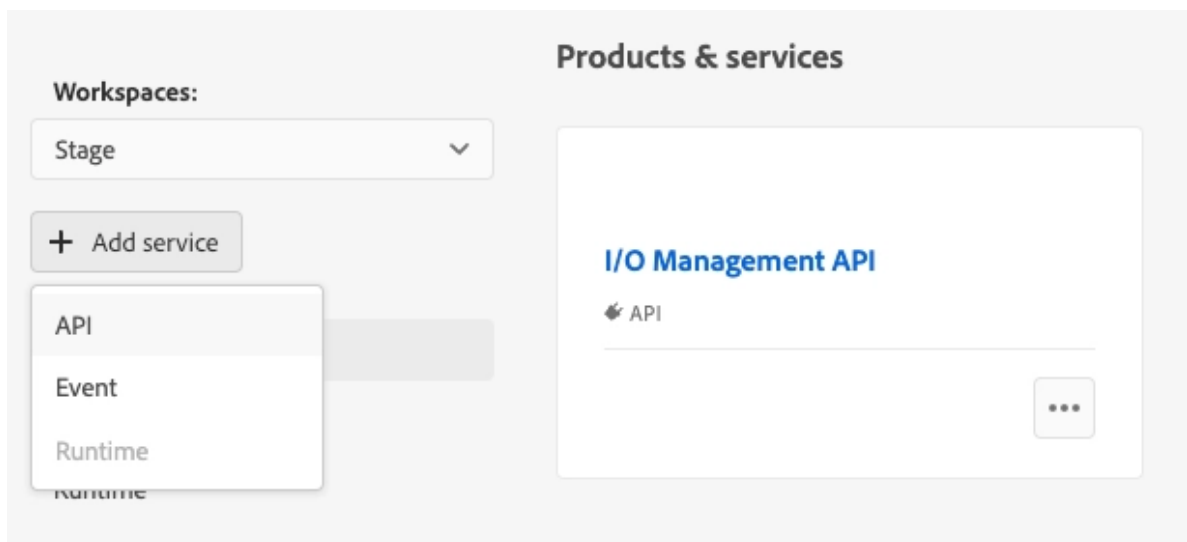
1

of 1

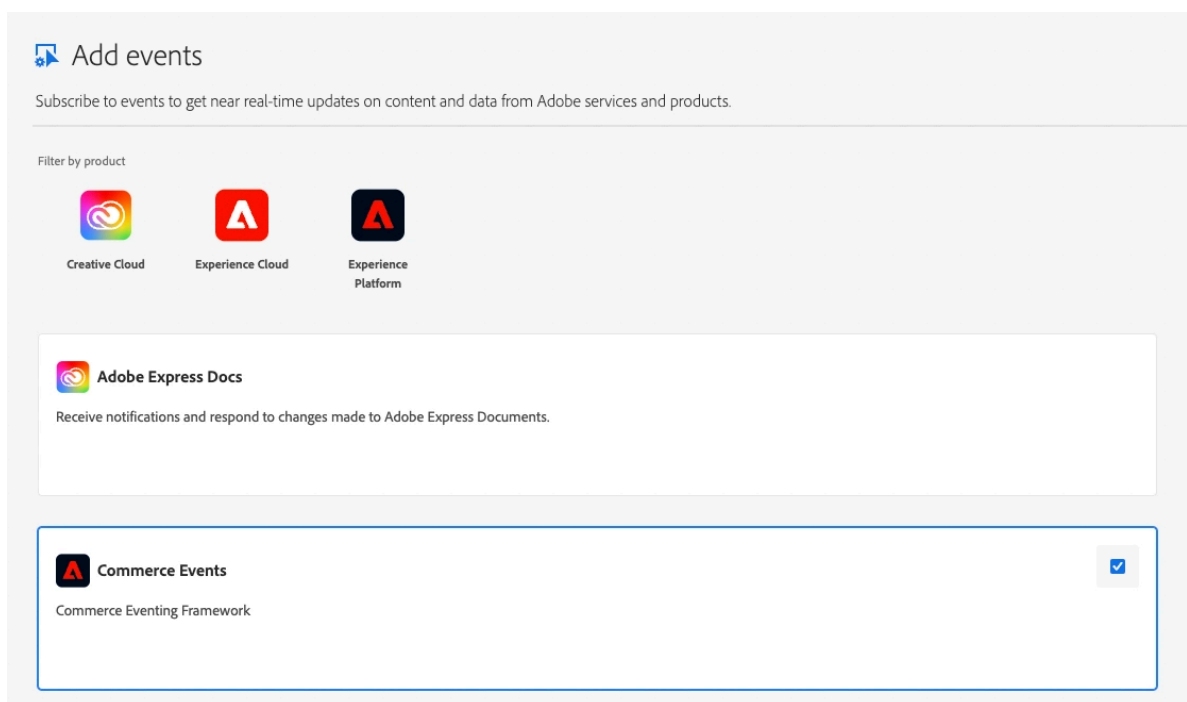
>

Event Name	Fields	Parent	Rules	Priority	Provider	Enabled	Action
observer.checkout_oope.checkout_submit_all_after	*	observer.checkout_submit_all_after	N/A	false		true	Select ▼

10. Return to your app environment. Click the **Add service** pop-up menu and select **Event**.

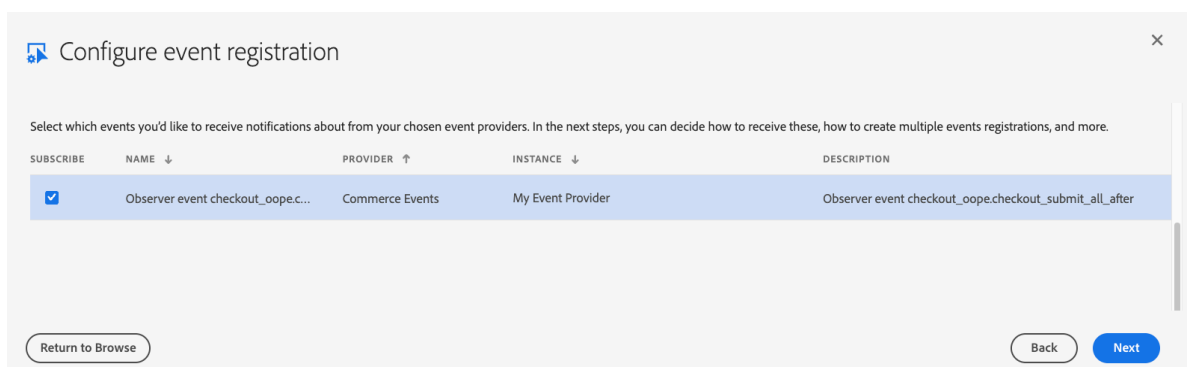


11. On the **Add events** page, select **Commerce events**. Then click **Next**.



12. On the **Configure event registration** page, select your event provider. Then click **Next**.

13. Select the events to subscribe to. Then click **Next**.



14. Update the **Event registration** name and **Event registration description** fields, for example, you can use **Order placed event**.
15. You will be asked how you would like to receive events. Select **Runtime action**, and then select the **stripe/validate-payment** action.

Configure event registration

Event Providers

- Commerce Events

Instances

- Adobe Commerce Events - Brand Two
DX_COMMERCE_EVENTS

Subscribed Events

Credentials

Event registration details

How to receive events

After the registration is saved, it will have an auto-generated Journaling URL. You also have the option to add a webhook URL, Runtime action, or Amazon EventBridge integration to receive notifications about events.

Journaling
Adobe's Journaling API collects all events for a specific time period. A unique endpoint will be generated after this registration is saved.
[Learn more.](#)

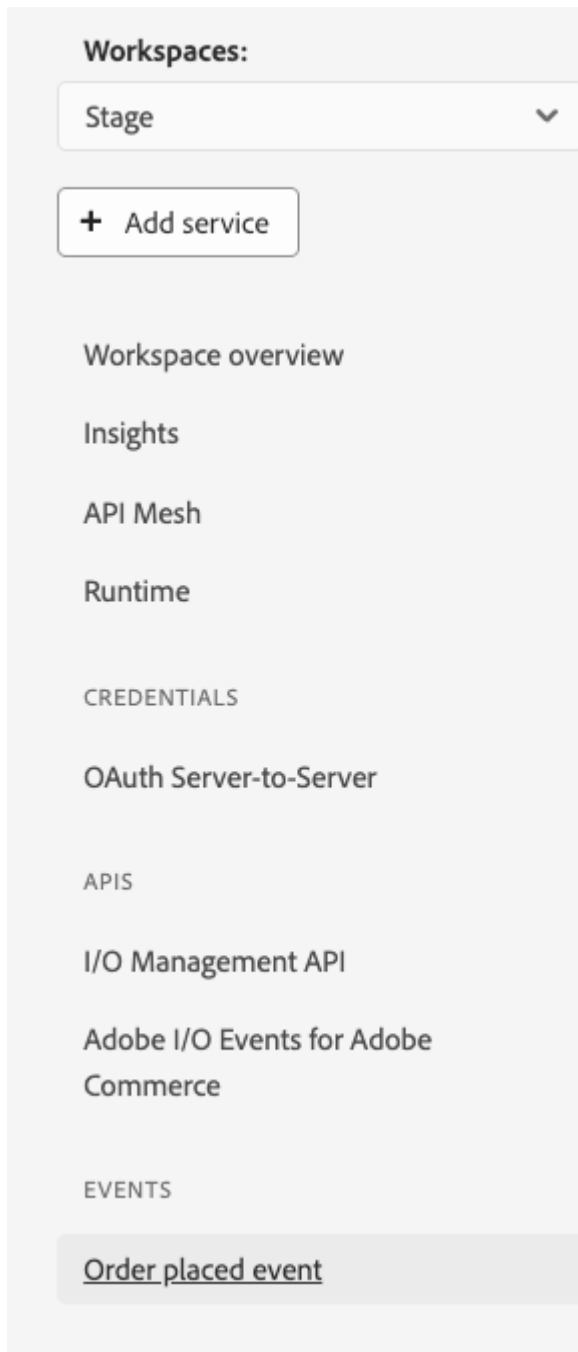
Add an optional way to receive events
You can add a webhook URL, Runtime action, or Amazon EventBridge integration to receive notifications about events.

OPTION 1
Webhook
Register a webhook by specifying a webhook URL and how often you'd like to receive the events. Each event will result in a HTTP request to this URL, notifying your application

OPTION 2
Runtime action
Select a Runtime action for where to consume your events. If you don't have Runtime set up, [view our documentation](#) to learn more about Runtime and how to deploy your first action.

OPTION 3
Amazon EventBridge
Add your AWS account ID and select AWS region to receive events directly from your AWS account. Once this registration is created, you will have **7 days** to navigate to your AWS account and configure the corresponding event bus to start receiving events. [View AWS documentation](#) for more information.

16. Click **Save configured events**.
17. The new event subscription will appear in the app's environment. You are now ready to start receiving events.



Add Stripe to your EDS storefront

You can add the Stripe payment method at the checkout page of your EDS storefront by utilizing a pre-built EDS block, which includes all functionality needed to render the payment form, collect the payment and place the order.

You can find this EDS block at <https://github.com/stripe/stripe-magento2-releases/tree/master/app-builder/blocks/stripe-payment>. Instructions on how to integrate it with your EDS storefront can be found inside the README.md file at the same location.