

Adobe Commerce integration starter kit

Partner Day 2024

Table of Contents

ADOBE COMMERCE INTEGRATION STARTER KIT	1
PARTNER DAY 2024	1
LAB OVERVIEW	3
LAB GOAL	3
KEY TAKEAWAYS	3
MEANINGFUL INFORMATION	3
ADOBE COMMERCE INTEGRATION STARTER KIT	5
REFERENCES FOR STARTER KIT	5
LESSON 1 – DOWNLOAD AND CONFIGURE STARTER KIT	6
LESSON 2 – ONBOARDING	14
LESSON 3 – SEND INFORMATION FROM ADOBE COMMERCE TO THE BACKOFFICE SYSTEM	17
LESSON 4 – SEND INFORMATION FROM A BACKOFFICE SYSTEM TO ADOBE COMMERCE	21
ADDITIONAL RESOURCES	25

Lab Overview

In this hands-on lab, you will learn how to start to use the Adobe Commerce integration starter kit for building your backend integrations.

Adobe Commerce starter kit enables developers with a scaffolding to build key integrations with 3rd party backoffice systems like a CRM, ERP, OMS, PIM or others.

This accelerator empowers solution partners to build their integration using a pre-defined integration architecture that follow Adobe best practices. The starter kit comes out of the box with pre-integrations for the most common data objects (product, customer, stock, order & shipment).

Lab Goal

This lab provides a simplified approach to understanding and using the Adobe Commerce starter kit. It showcases a bi-directional synchronization for one of the data objects.

During this lab, we would like to integrate with a backend system. All the examples provided assume that the business objective is to have a bi-directional synchronization between the eCommerce platform and a CRM.

The typical use case is that we allow shoppers to register directly on the website and by calling the customer care center, where a customer service agent can create an account on behalf of the shopper.

Key Takeaways

- Get familiar with using the Adobe Commerce starter kit
- Get familiar with the accelerator code structure
- Understand the onboarding scripts and get familiar in how to configure your desired data flow
- Connect Adobe Commerce with third party Backoffice systems and learn how to extend it for fit your merchant use cases
- Connect Backoffice system with Adobe Commerce and learn how to extend it for fit your merchant use cases

Meaningful information

- Adobe Commerce instance

- Admin URL: <https://partnerdays-jz4bb6q-zekho3zpsohow.us-4.magentosite.cloud/admin>
- Admin credentials: partnerdays / Adobe1234
- Developer IMS org: Commerce Extensibility Lab
- Github
 - Branch used for Lab <https://github.com/adobe/commerce-integration-starter-kit/tree/partnerday-lab-start>
 - Branch with all the code completed of the Lab <https://github.com/adobe/commerce-integration-starter-kit/tree/partnerday-lab-end>

Adobe Commerce integration starter kit

Integrating Adobe Commerce with your enterprise resource planning (ERP), order management system (OMS), or customer relationship management (CRM) is now easier with Adobe Commerce integration starter kit. The starter kit reduces the cost of integrating with ERP solutions and improves reliability of real-time connections. It includes reference integrations for commonly used data objects like orders, products, and customers. It also includes onboarding scripts and a standardized architecture for developers to build while following best practices.

References for starter kit

Explore the structure: <https://developer.adobe.com/commerce/extensibility/starter-kit/structure/>

Lesson 1 – Download and configure Starter kit

💡 This lab uses the AIO CLI tool. Follow the instructions at [CLI install](#) to set it up.

Main GitHub repository for Starter Kit: <https://github.com/adobe/commerce-integration-starter-kit/>

For this Lab, the starting point will be the **partnerday-lab-start** branch.

If you want to jump straight to the solution, check out the **partnerday-lab-end** branch directly.

1. Clone the repo using SSH & Select the branch for partner day lab:

```
git clone https://github.com/adobe/commerce-integration-starter-kit.git
cd commerce-integration-starter-kit
git checkout partnerday-lab-start
```

2. Copy the `env.dist` file as `.env`

```
cp env.dist .env
```

3. Login the Adobe Developer Console at <https://developer.adobe.com/console/> selecting the **Commerce Extensibility Lab** organization.

Developer Console is the gateway to access all the services and tooling available as part of the Adobe Developer Ecosystem, including App Builder.

Use the username format:

user<**your seat number**>@adobeeventlab.com

You will use the same username for all labs.

Sign in

New user? [Create an account](#)


Email address

user@adobeeventlab.com

Continue

Login with the password provided – Adobe2024!

Enter your password



user@adobeeventlab.com
Email address

Password

Adobe2024!

☒ Stay signed in


Continue

Skip adding your mobile number if prompted by selecting “Not now”.

Add a mobile phone number

Your phone number can make it easier to access your account securely. Additionally, you can receive SMS notifications about billing, security and other important updates.

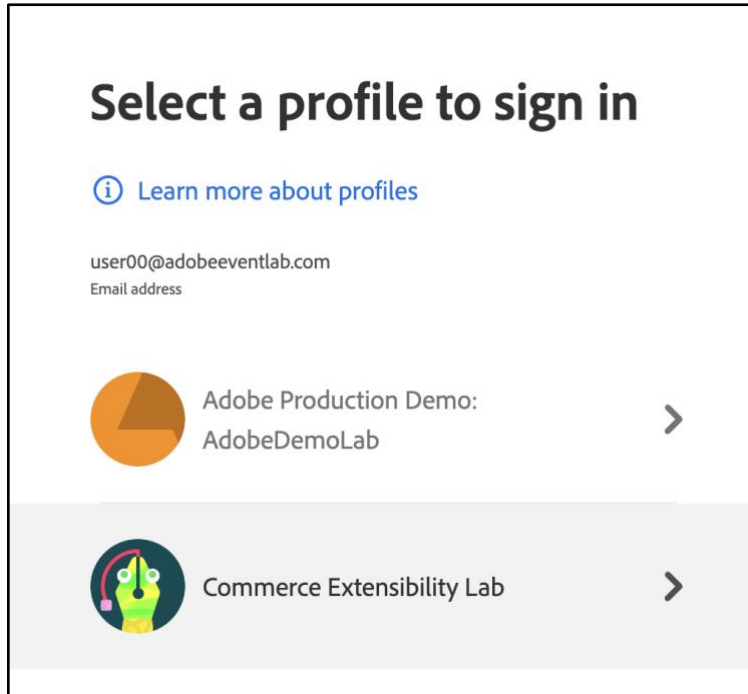
Phone number

 +1

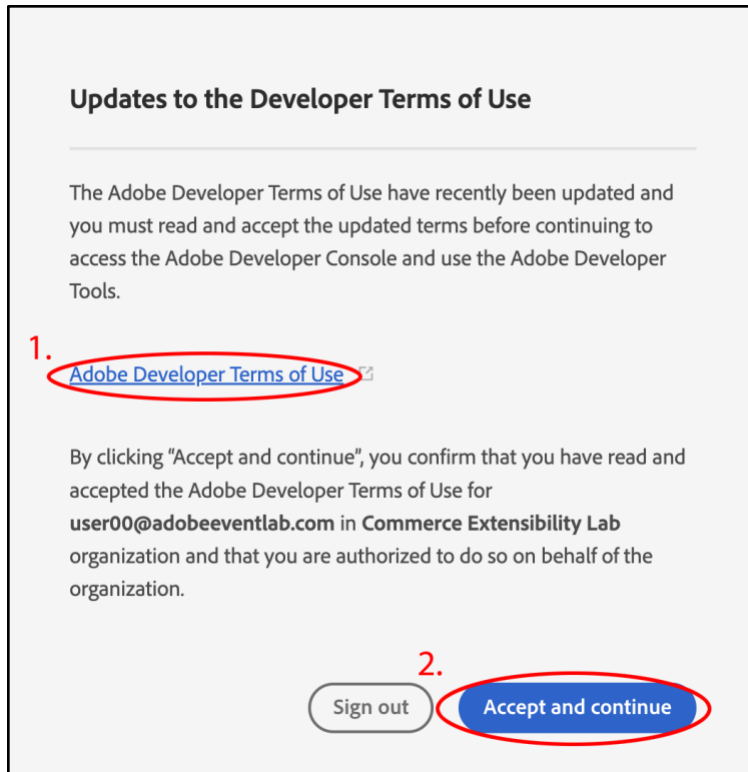
Not now

Continue

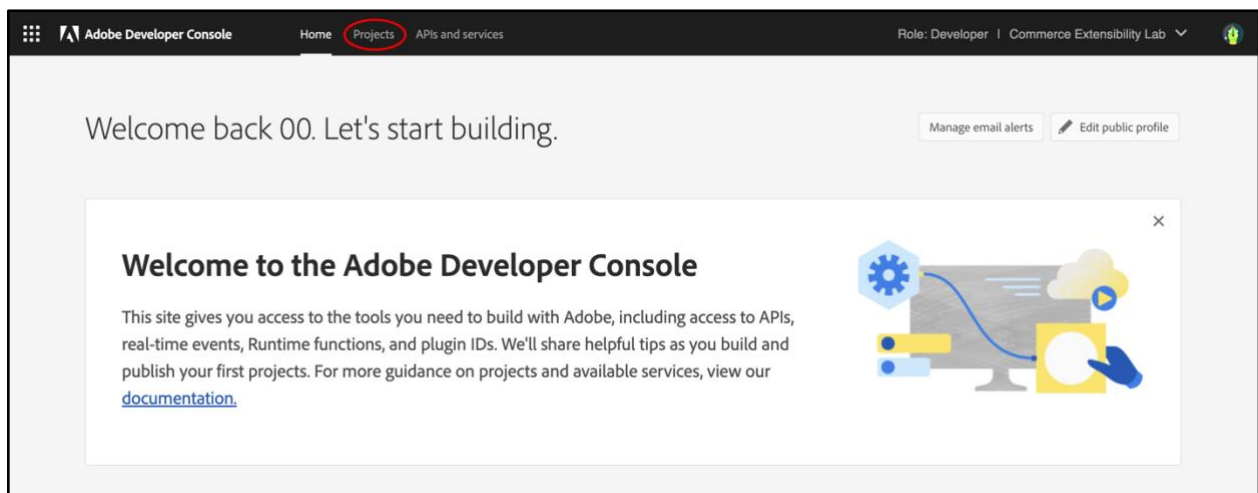
Select the “**Commerce Extensibility Lab**” profile to continue.



Accept the Terms of Use by clicking on the Blue link to open the “Adobe Developer Terms of Use” and then clicking on “Accept and Continue”



Once Adobe Developer Console has opened, select “projects” from the top menu. You are now using the Developer Console. Both API Mesh and I/O Events are supported by the UI and CLI.

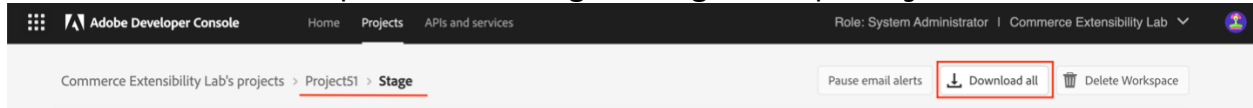


Select the project assigned to your user “Project<your seat number>”.

💡 *Tip – Use search to find your project*

a. Select your project

- b. Select the Stage workspace
- c. Download the workspace JSON configuration file by clicking on the 'Download all' button on the upper right corner of the page. Save it as "workspace.json" at commerce-integration-starter-kit/scripts/onboarding/config/workspace.json



4. Navigate to the directory where you cloned the starter kit github repository on your computer. Configure the .env file with all the information required. Please find below a table with more detailed information about where to find it.

This screenshot shows a code editor with the .env file. A red box highlights the following lines:

```

# OAuth client ID
OAUTH_CLIENT_ID=...
# OAuth client secret
OAUTH_CLIENT_SECRET=...

```

This screenshot shows the 'Generate access token' page in the Adobe Developer Console. The 'OAuth 2.0' tab is selected, and the 'Generate access token' button is highlighted with a red box.

This screenshot shows the 'Edit OAuth integration' page in the Adobe Developer Console. The 'Integration Details' section is visible, showing fields for 'Consumer Key', 'Consumer Secret', 'Access Token', and 'Access Token Secret'.

This screenshot shows the 'Edit OAuth integration' page in the Adobe Developer Console. The 'Current User Identity Verification' section is visible, showing fields for 'User Name' and 'User Password'.

Inside your project & workspace, click on the left menu on the OAuth Server-to-Server option.

These values can be copied from the Integration Details under System >

[illegible]

```

> aio console org select
  Getting Organizations...(node:7926) ExperimentalWarning: The Fetch API is an experimental feature. This feature could change at any time
  (Use 'node --trace-warnings ...' to show where the warning was created)
? Select Org: Commerce Extensibility Lab
Org selected Commerce Extensibility Lab
You are currently in:
1. Org: Commerce Extensibility Lab
2. Project: <no project selected>
3. Workspace: <no workspace selected>
> aio console project select
  Getting Projects...(node:8004) ExperimentalWarning: The Fetch API is an experimental feature. This feature could change at any time
  (Use 'node --trace-warnings ...' to show where the warning was created)
? Select Project: Project100
Project selected : Project100
You are currently in:
1. Org: Commerce Extensibility Lab
2. Project: Project100
3. Workspace: <no workspace selected>
> aio console workspace select
  Getting Workspaces...(node:8062) ExperimentalWarning: The Fetch API is an experimental feature. This feature could change at any time
  (Use 'node --trace-warnings ...' to show where the warning was created)
? Select Workspace: Stage
Workspace selected Stage
You are currently in:
1. Org: Commerce Extensibility Lab
2. Project: Project100
3. Workspace: Stage
> aio app use
You are currently in:
1. Org: <no org selected>
2. Project: <no project selected>
3. Workspace: <no workspace selected>

? Switch to a new Adobe Developer Console configuration: A. Use the global Org / Project / Workspace configuration:
  1. Org: Commerce Extensibility Lab
  2. Project: 344WheatMuskox
  3. Workspace: Stage
  Retrieving services supported by the Organization...(node:8245) ExperimentalWarning: The Fetch API is an experimental feature. This feature could change at any time
  (Use 'node --trace-warnings ...' to show where the warning was created)
? The file /Users/rojo/repos/partnerdays/partner100/commerce-integration-starter-kit/.env already exists: Merge

✓ Successfully imported configuration for:
1. Org: Commerce Extensibility Lab
2. Project: 344WheatMuskox
3. Workspace: Stage.

```

6. Install and deploy application

```

| npm install
| aio app deploy

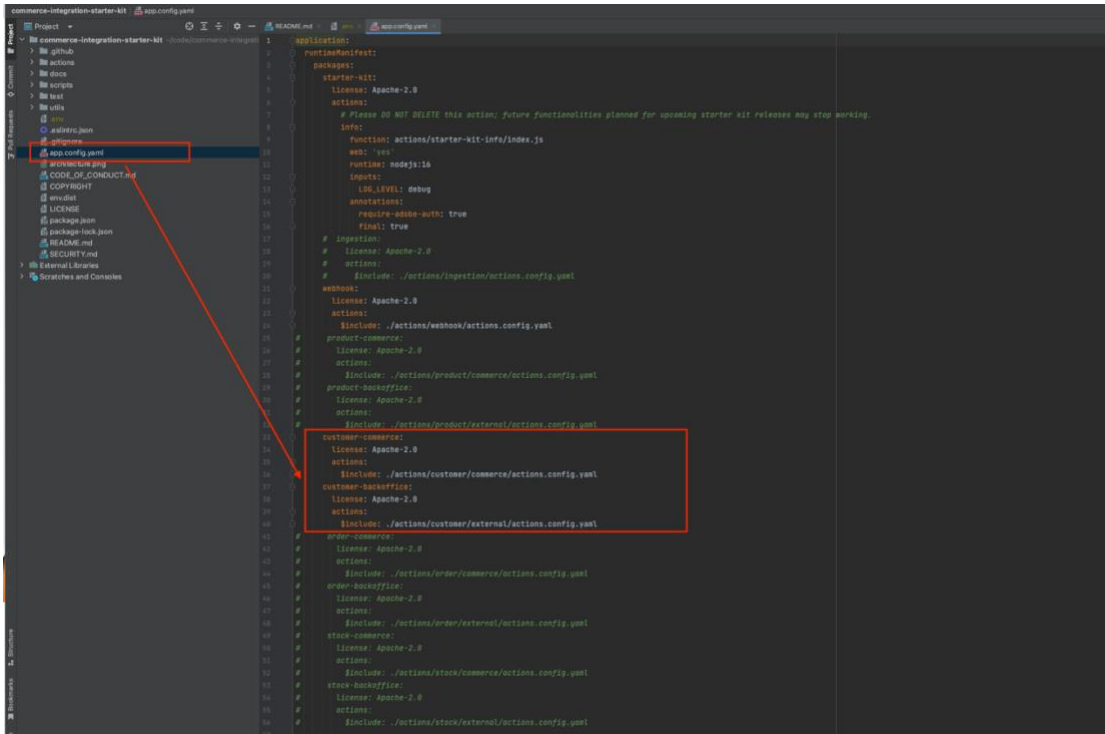
```

```

> aio app deploy
✓ Built 15 action(s) for 'application'
  no frontend or a build already exists, skipping frontend build for 'application'
✓ Deployed 17 action(s) for 'application'
  no frontend, skipping frontend deploy 'application'
Your deployed actions:
web actions:
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/web/starter-kit/info
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/web/webhook/check-stock
non-web actions:
  -> starter-kit/_secured_info
  -> webhook/_secured_check-stock
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/customer-commerce/consumer
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/customer-commerce/created
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/customer-commerce/updated
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/customer-commerce/deleted
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/customer-commerce/group-updated
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/customer-commerce/group-deleted
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/customer-backoffice/consumer
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/customer-backoffice/created
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/customer-backoffice/updated
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/customer-backoffice/deleted
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/customer-backoffice/group-created
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/customer-backoffice/group-updated
  -> https://1340225-586amethystlton-stage.adobeioruntime.net/api/v1/customer-backoffice/group-deleted
skipping publish phase...
Successful deployment 🚀

```

To match the business objective and speed up the development cycle, we defined in the app.config.yaml that only customer data flow will be deployed.



7. Check in developer.adobe.com that the runtime actions are now available.

Home

Projects

APIs and services

Role: System Administrator

Commerce Extensibility Lab

Commerce Extensibility Lab's projects

Project100

Stage

Runtime

Documentation

LAST REFRESH JUNE 8, 2024 12:13 PM

Refresh

Delete Runtime

Project overview

Activity Log

Workspaces:

Choose a workspace...

+ Add service

Workspace overview

Insights

Runtime

CREDENTIALS

OAuth Server-to-Server

API Key

APIs

I/O Management API

I/O Events

Adobe I/O Events for Adobe Commerce

API Mesh for Adobe Developer App Builder

RUNTIME NAMESPACE

3117813-344wheatmuskow-stage

Copy

Get started with Adobe I/O Runtime

There are a few steps to take outside of the Developer Console in order to start using Adobe I/O Runtime. [Learn more](#)

Set up your environment

You'll have to install and configure a couple of tools on your machine before you're able to create and run actions.

User Defined Actions

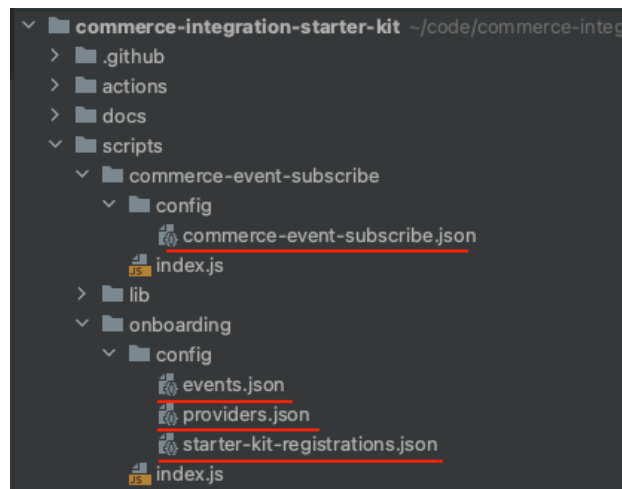
7 ACTIONS

NAME	URL	TIMEOUT	MEMORY LIMIT	VERSION
group-deleted	https://runtime.adobe.io/api/v1/namespaces/3117813-344wheatmuskow-stage/actions/customer-backoffice/group-deleted	60000 ms	256 MB	0.01
group-updated	https://runtime.adobe.io/api/v1/namespaces/3117813-344wheatmuskow-stage/actions/customer-backoffice/group-updated	60000 ms	256 MB	0.01
group-created	https://runtime.adobe.io/api/v1/namespaces/3117813-344wheatmuskow-stage/actions/customer-backoffice/group-created	60000 ms	256 MB	0.01
deleted	https://runtime.adobe.io/api/v1/namespaces/3117813-344wheatmuskow-stage/actions/customer-backoffice/deleted	60000 ms	256 MB	0.01
updated	https://runtime.adobe.io/api/v1/namespaces/3117813-344wheatmuskow-stage/actions/customer-backoffice/updated	60000 ms	256 MB	0.01
created	https://runtime.adobe.io/api/v1/namespaces/3117813-344wheatmuskow-stage/actions/customer-backoffice/created	60000 ms	256 MB	0.01
consumer	https://runtime.adobe.io/api/v1/namespaces/3117813-344wheatmuskow-stage/actions/customer-backoffice/consumer	60000 ms	256 MB	0.01

Lesson 2 – Onboarding

The starter kit provides out-of-the-box a standard set and onboarding scripts to simplify the most common task in an event-based architecture. The main objective of that runtime action is that during the installation of the package that we are building, there are low/no manual activities required.

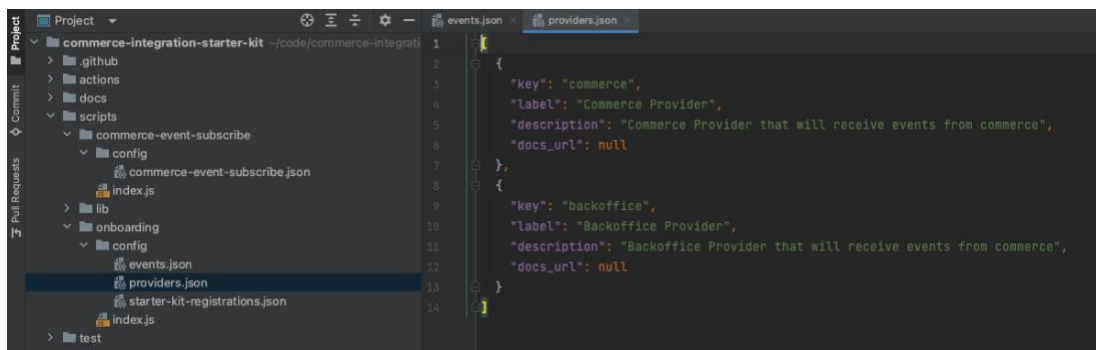
To ensure flexibility for the onboarding scripts there few JSON configuration files for being able



To ensure flexibility for the onboarding scripts, the starter kit code provides a few different JSON configuration files to match your business requirements. All the files were pre-configured for that lab to match the requirements defined during the Lab Goal section.

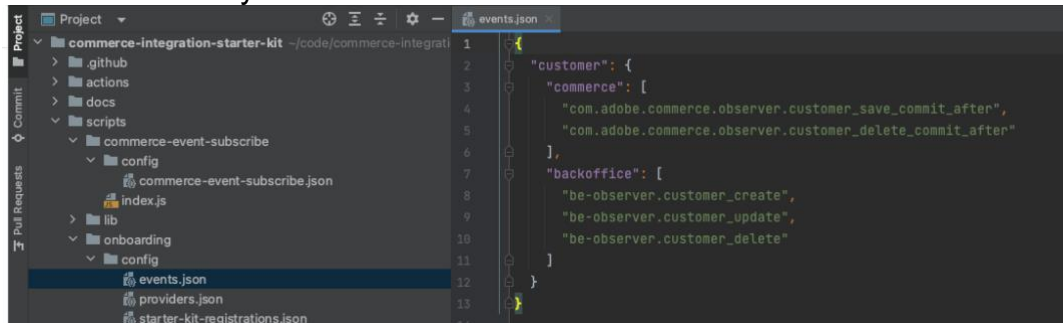
providers.json

Define the providers required during the integration. The "key" field will reference the providers in other configuration files. The rest of the fields are descriptive.



events.json

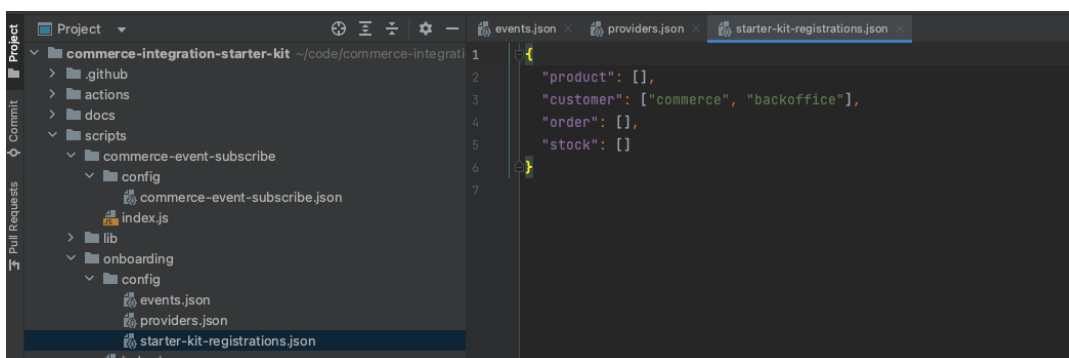
Define all the meaningful events that integration needs to care about. If more events are required, add them to the correct provider (provider key defined in the providers.json) and the data flow that handles them. For example, if you want to send a customer group event from Commerce > CRM, you should add that event inside the "customer" > "commerce" event array.



```
1  {
2    "customer": {
3      "commerce": [
4        "com.adobe.commerce.observer.customer_save_commit_after",
5        "com.adobe.commerce.observer.customer_delete_commit_after"
6      ],
7      "backoffice": [
8        "be-observer.customer_create",
9        "be-observer.customer_update",
10       "be-observer.customer_delete"
11      ]
12    }
13  }
```

starter-kit-registrations.json

Defines the registration required to be defined. Registration in App Builder defines the runtime action that must be triggered when a specific event reaches the provider.



```
1  {
2    "product": [],
3    "customer": ["commerce", "backoffice"],
4    "order": [],
5    "stock": []
6  }
```

commerce-event-subscribe.json

Defines the events that must be subscribed to Adobe Commerce and the required fields. That commerce-event-subscribe command is only applicable once the Eventing module is 1.6.0+ (<https://developer.adobe.com/commerce/extensibility/events/release-notes/#version-160>)

Now is time to realize the onboarding, go to your terminal and run:

```
| npm run onboard
```


Project overview

Activity Log

MODIFIED

Jun 6, 2024
By Angel Rojo
Marjuan

CREATED

Jun 6, 2024
By Angel Rojo
Marjuan

LAST SERVICE ACTIVITY ⓘ

Jun 7, 2024

DESCRIPTION

No description

EMAIL ALERTS STATUS ⓘ

Active

Workspaces:

Stage

+ Add service

Workspace overview

Insights

Runtime

CREDENTIALS

OAuth Server-to-Server

APIS

I/O Management API

I/O Events

Adobe I/O Events for Adobe Commerce

EVENTS

Backoffice Customer Synchronization

Commerce Customer Synchronization

Products & services

I/O Management API

API

...

I/O Events

API

...

Adobe I/O Events for Adobe Com...

API

...

Commerce Customer Synchroniza...

EVENTS

...

Backoffice Customer Synchronizat...

EVENTS

...

1340225-partnerdays-stage

RUNTIME

11 USER DEFINED ACTIONS

Credentials

OAuth Server-to-Server

CREDENTIAL

API KEY (CLIENT ID)

a0b8a6ebc01a45daa8ab173cb6e03872

Copy

Generate access token

Project overview

Activity Log

Registration Details

Debug Tracing

Event Browser

EVENT REGISTRATION DESCRIPTION

Commerce Customer Synchronization

Workspaces:

Stage

+ Add service

Workspace overview

Insights

Runtime

CREDENTIALS

OAuth Server-to-Server

APIS

I/O Management API

I/O Events

Adobe I/O Events for Adobe Commerce

EVENTS

Backoffice Customer Synchronization

Commerce Customer Synchronization

Event Delivery Method

RUNTIME ACTION

customer-commerce/consumer - 1340225-partnerdays-stage

JOURNALING UNIQUE API ENDPOINT

https://events-vaa6.adobe.io/events/organizations/1340225/integrations/629219/2dc411a1-be06-45e9-a299-5e9b1b93cae3

Copy

Event Providers

+ Add event providers

Commerce Provider - partnerdays-stage

View docs

Subscribed Events

EVENT NAME	EVENT PROVIDER		
Observer event customer_save_commit_after	Commerce Provider - par...	Copy	Remove
Observer event customer_delete_commit_after	Commerce Provider - par...	Copy	Remove

< 1 of 1 pages >

Connected credentials

OAuth Server-to-Server

CREDENTIAL

API KEY (CLIENT ID)

a0b8a6ebc01a45daa8ab173cb6e03872

Copy

Generate access token

Lesson 3 – Send information from Adobe Commerce to the Backoffice system

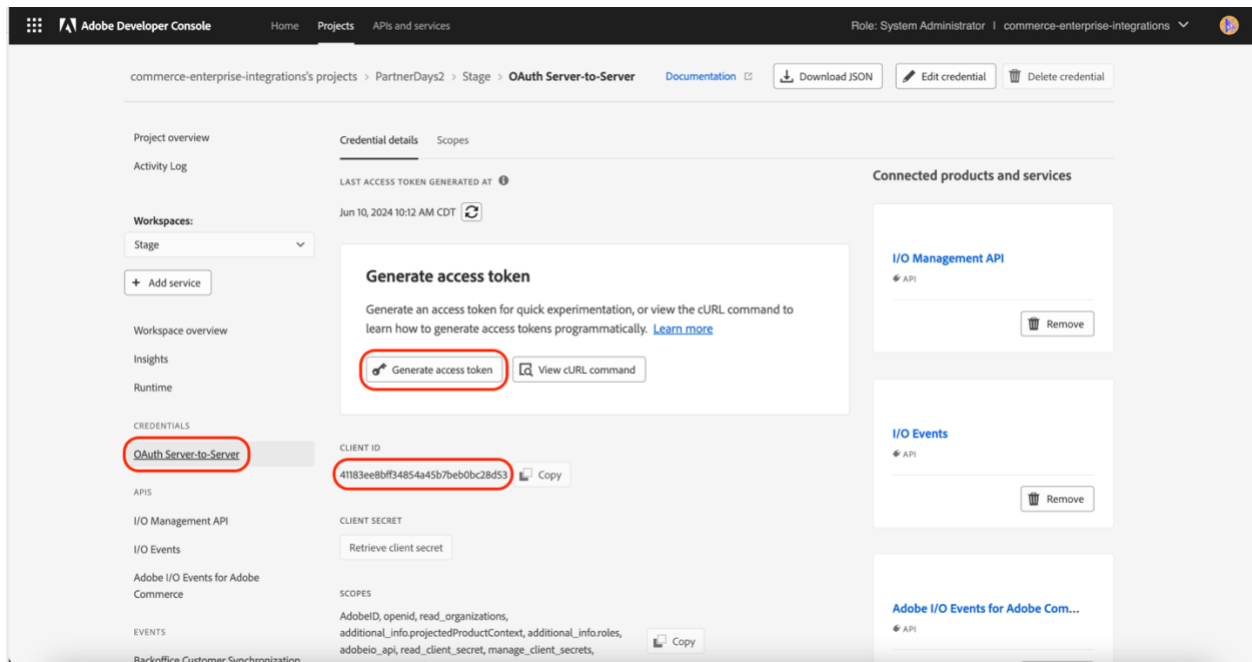
We will be connecting with a mock CRM at <https://1339710-967summiteventing-stage.adobeio-static.net/index.html#/>

Simulate a customer being created in Commerce by executing the following curl request in a terminal after replacing the placeholders (they are formatted as <ABCD>)

```
curl --request POST \
--url https://eventsingress.adobe.io/ \
--header 'Authorization: Bearer <ACCESS_TOKEN>' \
--header 'Content-Type: application/cloudevents+json' \
--header 'x-api-key: <CLIENT_ID>' \
--data '{
  "datacontenttype": "application/json",
  "specversion": "1.0",
  "source": "urn:uuid:<COMMERCE_EVENT_PROVIDER_ID>",
  "type": "com.adobe.commerce.observer.customer_save_commit_after",
  "id": "<UUID_v4>",
  "data": {
    "value": {
      "id": 435,
      "email": "<YOUR_EMAIL>",
      "firstname": "<YOUR_NAME>",
      "lastname": "<YOUR_LASTNAME>",
      "created_at": "2024-06-06 15:38:03",
      "updated_at": "2024-06-06 15:38:03"
    }
  }
}'
```

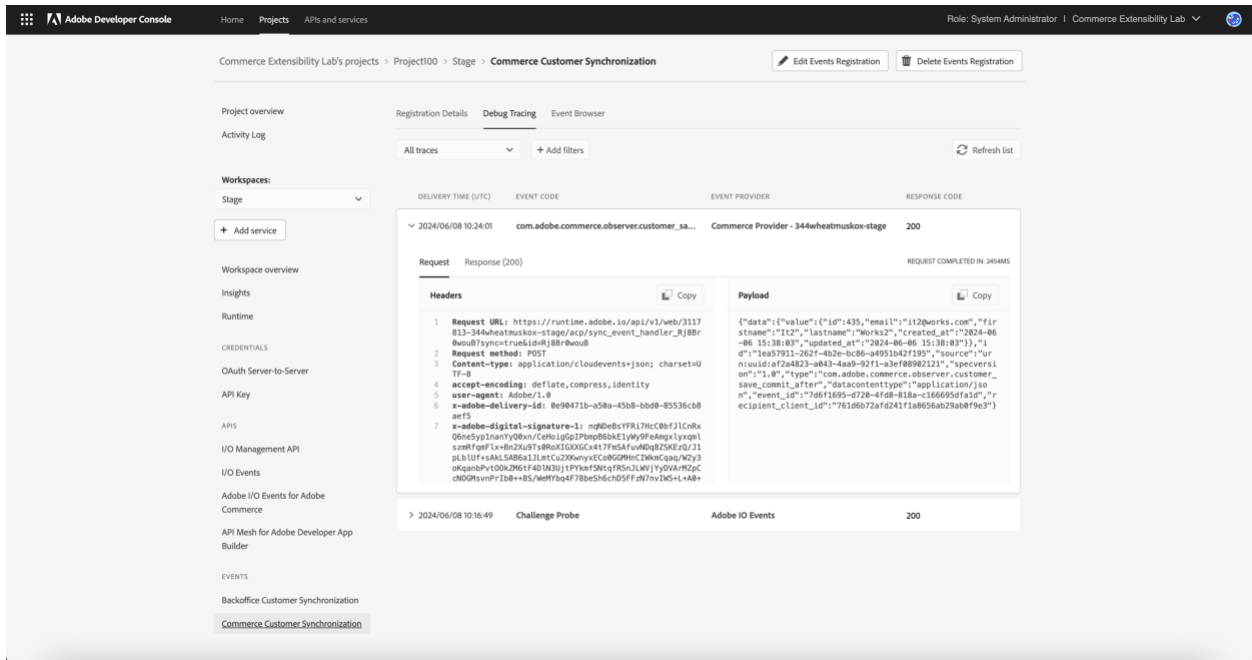
(if you use Insomnia, the collection at [partnerdays-postman.yaml](#) contains a “Commerce customer created” request equivalent to the above curl command)

An ACCESS_TOKEN can be generated in the OAuth Server-to-Server section of your workspace in the Adobe Developer Console. The CLIENT_ID can also be taken from that page.

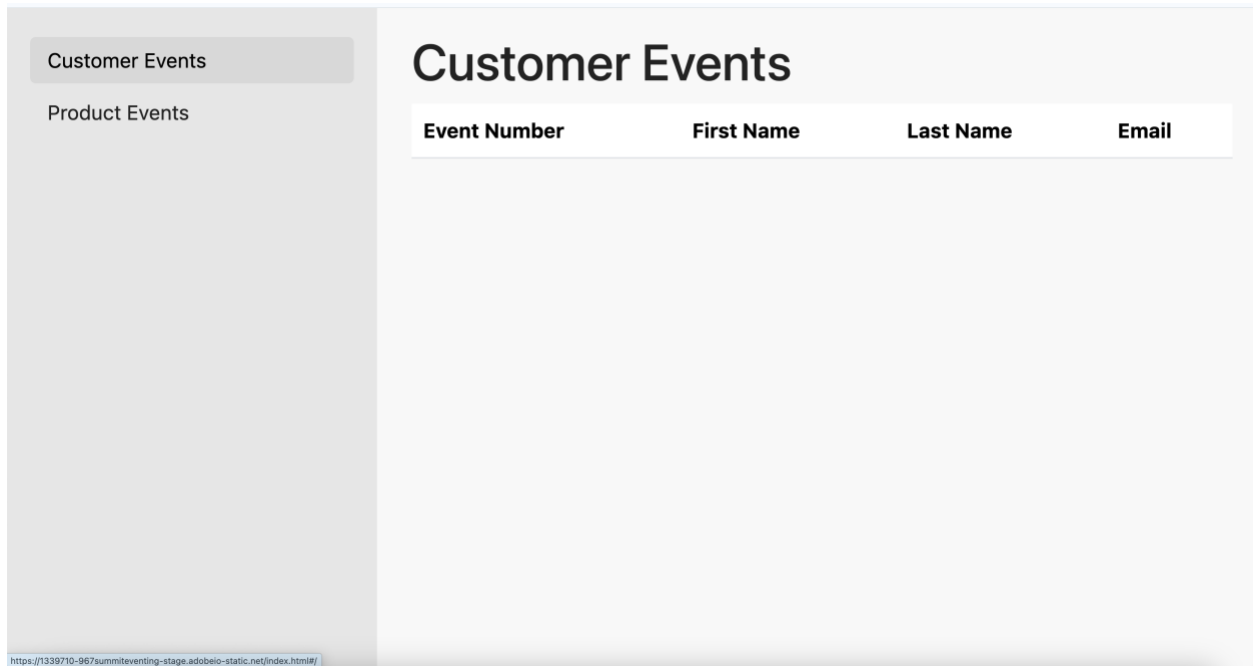


Confirm that an event has been handled by the “Commerce Customer Synchronization” subscription in the Developer Console.

Select the “Commerce Customer Synchronization” event in the Project’s left navigation and go to the Debug Tracing tab.



Unsurprisingly the customer is not shown in the mock CRM UI. Why? Because the event received activates a customer-commerce/created runtime action that does not forward the request to the mock CRM.



In order to have the customer created in the mock CRM, make the following changes to the code in your project:

- Replace `actions/customer/commerce/created/transformer.js` with [transformer.js](#).
This code transforms the incoming Commerce event into a payload understandable by the mock CRM API
- Replace `actions/customer/commerce/created/sender.js` with [sender.js](#).
This code sends the transformed payload to the mock CRM API by using the client implemented in `actions/customer/mockapi-api-client.js`

After doing the above code changes, redeploy the new version of the runtime action by running

```
| aio app deploy --action=customer-commerce/created
```

Execute again the curl command to send the event to App Builder and confirm that a second event has been handled by the “Commerce Customer Synchronization” in the Developer Console.

Adobe Developer Console

Home Projects APIs and services

Role: System Administrator | Commerce Extensibility Lab

Commerce Extensibility Lab's projects > Project100 > Stage > Commerce Customer Synchronization

Edit Events Registration Delete Events Registration

Project overview Registration Details Debug Tracing Event Browser

Activity Log

All traces + Add filters Refresh list

Workspaces:

Stage + Add service

Workspace overview

Insights

Runtime

CREDENTIALS

OAuth Server-to-Server

API Key

APIS

I/O Management API

I/O Events

Adobe I/O Events for Adobe Commerce

API Mesh for Adobe Developer App Builder

EVENTS

Backoffice Customer Synchronization

Commerce Customer Synchronization

DELIVERY TIME (UTC) EVENT CODE EVENT PROVIDER RESPONSE CODE

2024/06/08 10:45:46 com.adobe.commerce.observer.customer_sa... Commerce Provider - 344wheatmuskox-stage 200

Request Response (200) REQUEST COMPLETED IN 369MS

Headers Payload

1 Request URL: https://runtime.adobe.io/api/v1/web/3117813-344wheatmuskox-stage/acp/sync_event_handler_RJ88r8wou8?sync=true&id=RJ88r8wou8

2 Request method: POST

3 Content-type: application/cloudevents+json; charset=UTF-8

4 accept-encoding: deflate,compress,identity

5 user-agent: Adobe/1.0

6 x-adobe-delivery-id: e80be99-7f37-43ff-bffe-e48911faf1f1

7 x-adobe-digital-signature-1: GCv02/Y0Uc1cFpW108yPkvtowf1DtrvY7Ev2g2ndwHkLdPh2L/vxly44168f13437J0D9v=5L+CKIuy0K5J0PFS6rC2UvecbFxc30ABgz827Hukq/QefrEqHe1k7JghqP75/n2Gy8gk08qWdCk4v0p3y42GR2ekCzd2BLu1Th2089q6LIT3u3KCTw6C3P2T56uL3>=xv1+PacHkV8qL3Y5q728rsQ15C0kXLo51PT8vdfUaL9W0/F9YAUR0QE3k1z+1nKj0s39y08f

{ "data": { "value": { "id": 435, "email": "it2@works.com", "first_name": "It2", "last_name": "Works2", "created_at": "2024-06-08 15:38:03", "updated_at": "2024-06-08 15:38:03", "id": "7a15a8c6-24f4-4321-828f-801f8057465", "source": "runtime:uid:f2a4823-a843-4a99-92f1-a3ef08982121", "specversion": "1.0", "type": "com.adobe.commerce.observer.customer_save_comml_after", "datacontenttype": "application/json", "event_id": "8812803a-a7b4-4943-948e-bd7158113ed6", "recipient_client_id": "761d6b72af0241f1a8656ab29aaf9e3" } } }

> 2024/06/08 10:24:01 com.adobe.commerce.observer.customer_sa... Commerce Provider - 344wheatmuskox-stage 200

> 2024/06/08 10:16:49 Challenge Probe Adobe IO Events 200

Also, confirm that this time the mock CRM received the request from our App Builder application by checking that the user details are shown on <https://1339710-967summiteventing-stage.adobeio-static.net/index.html#/>

Customer Events

Product Events

Customer Events

Event Number	First Name	Last Name	Email
1	It2	Works2	it2@works.com

Lesson 4 – Send information from a Backoffice system to Adobe Commerce

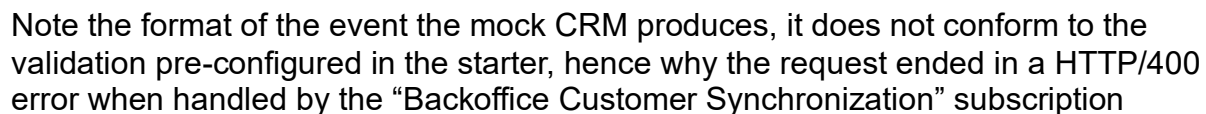
We will be connecting with a mock CRM at <https://1339710-967summiteventing-stage.adobeio-static.net/index.html#/>

Simulate a customer being created in the CRM by executing the following curl request in a terminal after replacing the placeholders (they are formatted as <ABCD>)

```
curl --request POST \
  --url https://eventsingress.adobe.io/ \
  --header 'Authorization: Bearer <ACCESS_TOKEN>' \
  --header 'Content-Type: application/cloudevents+json' \
  --header 'x-api-key: <CLIENT_ID>' \
  --data '{
    "datacontenttype": "application/json",
    "specversion": "1.0",
    "source": "urn:uuid:<BACKOFFICE_EVENT_PROVIDER_ID>",
    "type": "be-observer.customer_create",
    "id": "<UUID_V4>",
    "data": {
      "PersonFirstName": "<YOUR_NAME>",
      "PersonLastName": "<YOUR_LASTNAME>",
      "PersonMiddleName": "Middle",
      "Email": "<YOUR_EMAIL>",
      "CustomerGroupId": "30",
      "SalesCurrencyCode": "USD",
      "PrimaryContactPhone": "555-555-1234",
      "PartyType": "Person",
      "AddressStreet": "1 Test Street",
      "AddressCountryRegionISOCode": "US",
      "AddressZipCode": "00210",
      "AddressCity": "Portsmouth",
      "AddressState": "NH",
      "AddressLocationRoles": "Invoice",
      "AddressDescription": "Test Billing",
      "DeliveryAddressStreet": "2 Test Avenue",
      "DeliveryAddressCountryRegionISOCode": "US",
      "DeliveryAddressZipCode": "00210",
      "DeliveryAddressCity": "Portsmouth",
      "DeliveryAddressState": "NH",
      "DeliveryAddressDescription": "Test Delivery"
    }
  }'
```

(if you use Insomnia, the collection at [partnerdays-postman.yaml](#) contains a “CRM customer created” request equivalent to the above curl command)

An ACCESS_TOKEN can be generated in the OAuth Server-to-Server section of your workspace in the Adobe Developer Console.



This file makes the JSON schema for the incoming event conform to the payload we expect to receive from the mock CRM

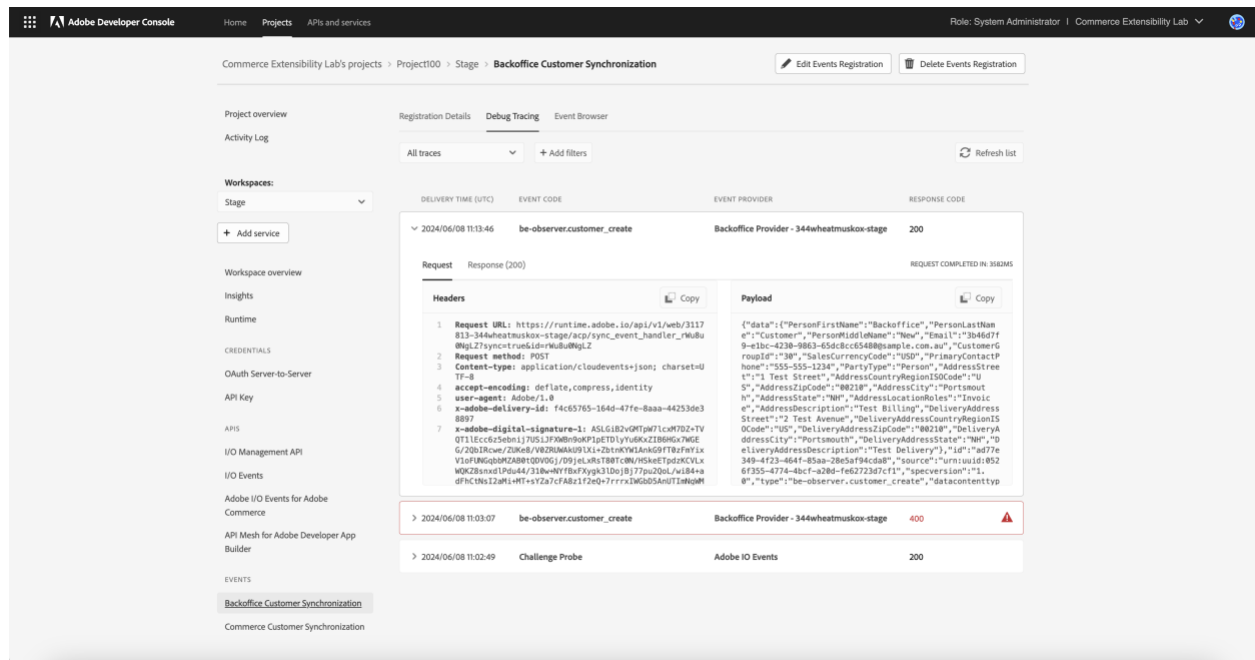
- Replace `actions/customer/external/created/transformer.js` with [transformer.js](#).

This code transforms the incoming event into the payload that Commerce expects to receive when creating a customer

After doing the above code changes, redeploy the new version of the runtime action by running

```
| aio app deploy --action=customer-backoffice/created
```

Execute again the curl command to send the event to App Builder, and confirm that this second event has been successfully handled by the “Backoffice Customer Synchronization” in the Developer Console.



The screenshot shows the Adobe Developer Console interface. The main panel displays the 'Backoffice Customer Synchronization' event registration details. The event was received at 2024/06/08 11:13:46 with a response code of 200. The payload is visible, showing customer creation details. The console also shows a list of events, including a 'Challenge Probe' event with a 200 response code.

DELIVERY TIME (UTC)	EVENT CODE	EVENT PROVIDER	RESPONSE CODE
2024/06/08 11:13:46	be-observer.customer_create	Backoffice Provider - 344wheatmuskox-stage	200
2024/06/08 11:03:07	be-observer.customer_create	Backoffice Provider - 344wheatmuskox-stage	400
2024/06/08 11:02:49	Challenge Probe	Adobe IO Events	200

Also, confirm that this time Commerce received the request from our App Builder application and created a customer by checking that a customer with the values used when simulating the event exists in <https://partnerdays-jz4bb6q-zekho3zpsohow.us-4.magentosite.cloud/admin>.

Dashboard

Sales

Catalog

Customers

Marketing

Content

Reports

Stores

System

Find Partners & Extensions

Backoffice Customer

Q

11

partnerdays

← Back

Login as Customer

Delete Customer

Reset

Create Order

Reset Password

Force Sign-In

Manage Shopping Cart

Save and Continue Edit

Save Customer

CUSTOMER INFORMATION

Customer View

Account Information

Addresses

Orders

Returns

Shopping cart

Wish List

Gift Registry

Store Credit

Newsletter

Billing Agreements

Product Reviews

Account Information

Zoura Account Number (Unique)

Associate to Website * Main Website ?

Group * General

☐ Disable Automatic Group Change Based on VAT ID

Name Prefix

First Name * Backoffice

Middle Name/Initial New

Last Name * Customer

Name Suffix

Email * 3b46d7f9-e1bc-4230-9863-65dc8cc65480@sample.com.au

Allow remote shopping assistance No

24

Additional Resources

The following resources will help you get to know the extensibility tooling offered by Adobe Commerce and start developing.

💡 You can also find this entire [list of resources in our public documentation](#).

How does App Builder work?

- [Introducing App Builder for Commerce](#) - A high-level business value proposition and ecosystem.
- [Use Cases](#) - a one-hour video on common App Builder use cases.
- [Sample App](#) - An internally developed sample app with best practices and references to an example GitHub repo.
- [Demo](#) - A short video demonstrating a functional integration between Commerce and Amazon Marketplace.

How can I get set up?

- [Getting access to App Builder](#) - Information on how to request trial access or purchase a license.
- [Onboarding developers and admins](#) - Information on how to set up your organization after getting access to App Builder.

How do I start developing?

- Develop custom applications with Adobe Developer App Builder
 - [App Builder overview](#) - An overview of the App Builder ecosystem. The App Builder documentation applies to all Adobe products that integrate with App Builder, including Adobe Commerce.
 - [Creating your first App Builder application](#) - A generalized tutorial on creating your first App Builder app. If you are new to App Builder, Adobe recommends reviewing the App Builder overview before beginning this tutorial.
- Orchestration with API Mesh for Adobe Developer App Builder
 - [Getting started](#) - Information on how to get started with API Mesh.
 - [Creating a mesh](#) - Developer documentation on creating a mesh.
 - [Create a mesh walkthrough](#) - Documentation that provides a guided walkthrough of basic API Mesh functionality.
 - [API Mesh video tutorials](#) - A set of video tutorials that demonstrate basic API Mesh functionality
- Build event-driven integrations with Adobe I/O Events for Adobe Commerce

- [Adobe I/O Events](#) - Developer documentation for general Adobe I/O Events functionality, not specific to Adobe Commerce. View this documentation for information about consuming events in App Builder using journaling, runtime actions, retry mechanisms, and more.
- [Adobe I/O Events for Adobe Commerce](#) - Developer documentation for Adobe I/O Events for Adobe Commerce.
- [Installation](#) - Developer docs for setting up the I/O Events for Adobe Commerce.
- [Eventing video tutorials](#) - A set of developer-facing video tutorials that demonstrate how to set up I/O Events for Adobe Commerce
- Extend the Admin UI with the Admin UI SDK
 - [Admin UI SDK overview](#) - Developer documentation on how to extend and customize the Commerce Admin to include custom menus and pages.
 - [Installation](#) - Developer documentation on how to set up the Admin UI SDK.
 - [Extension points](#) - Developer documentation on how to use existing extension points in any Adobe Developer App Builder application that customizes the Adobe Commerce Admin.
- Extend native Commerce processes with webhooks
 - [Overview](#) - An architectural overview of webhooks.
 - [Setup and installation](#) - Information on how to install Commerce modules necessary for implementing webhooks.
 - [Configuration](#) - Developer documentation on how to configure hooks for Adobe Commerce.
 - [Common use cases](#) - Descriptions of common Adobe Commerce use cases for webhooks.
 - [Webhooks](#) - Learn how to extend, override, and create hooks using the Adobe Commerce Admin.