Adobe Experience Platform Bootcamp Deep Dive Edition

DOCUMENTATION AND LABS

Contents

Lab 1 - Lookup Stores

Lab 2 Customer Accounts

Lab 3 - Historical

Adobe Experience Platform - Data Pipeline Stores

Adobe Experience Platform Bootcamp Deep Dive Edition

Name _____

Sandbox _____

1. Lab Overview

Ingest Stores data into AEP in batch mode using Data Landing Zone. This Lab will leverage delimited data on Landing Zone and ML Recommendations to map most of the source data. This job will be scheduled to run every 30 days.

This lab will introduce you to the basic data ingestion process.

Expected time: 20 minutes

2. Learning Objectives

What should you walk away with after taking this Lab?

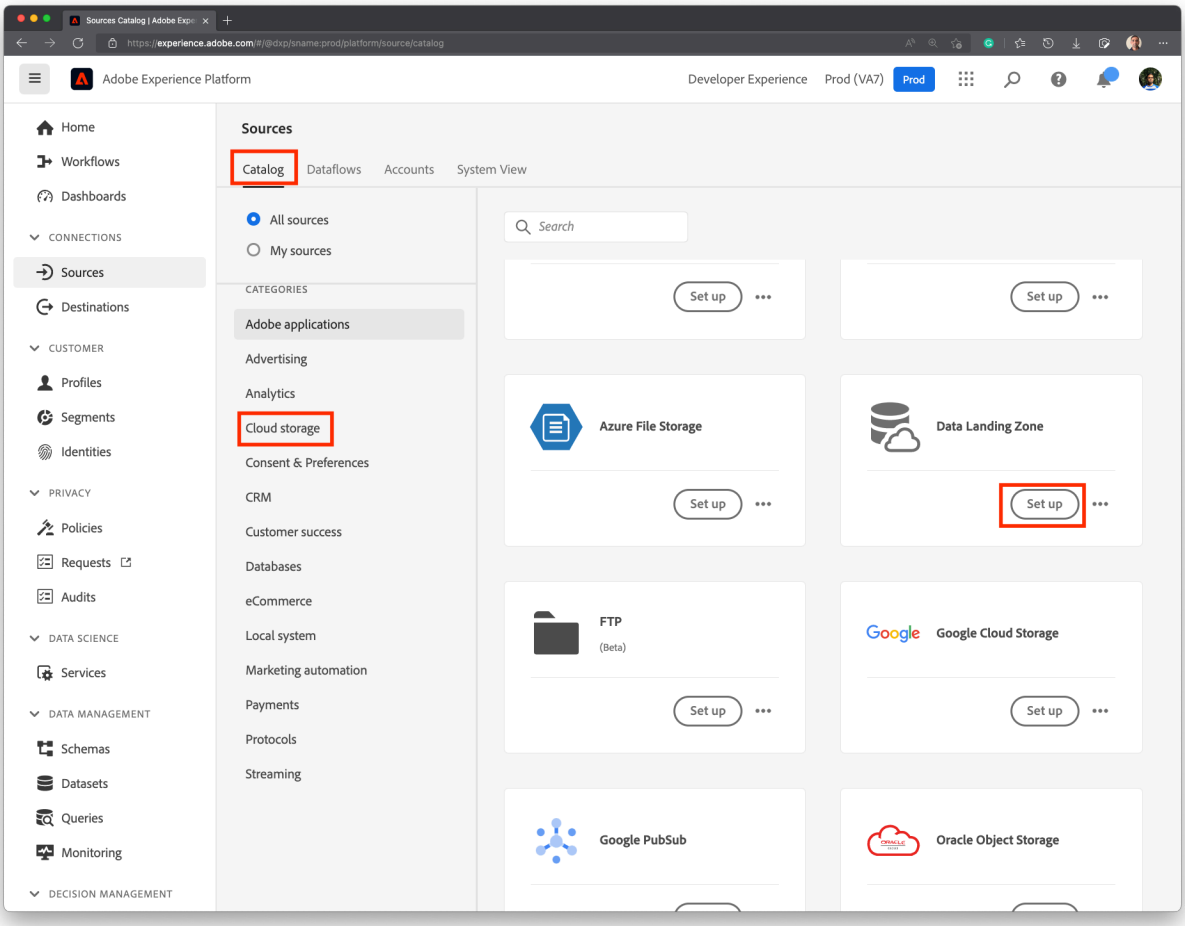
- Understand the basic data ingestion process
- Use Data Landing Zone as a source
- Use Data Prep to Map the non-XDM data to XDM
- Scheduling batch workflows
- Preview the data in dataset

3. Lab Task – Creation of the Dataflow

3.1. Select source data

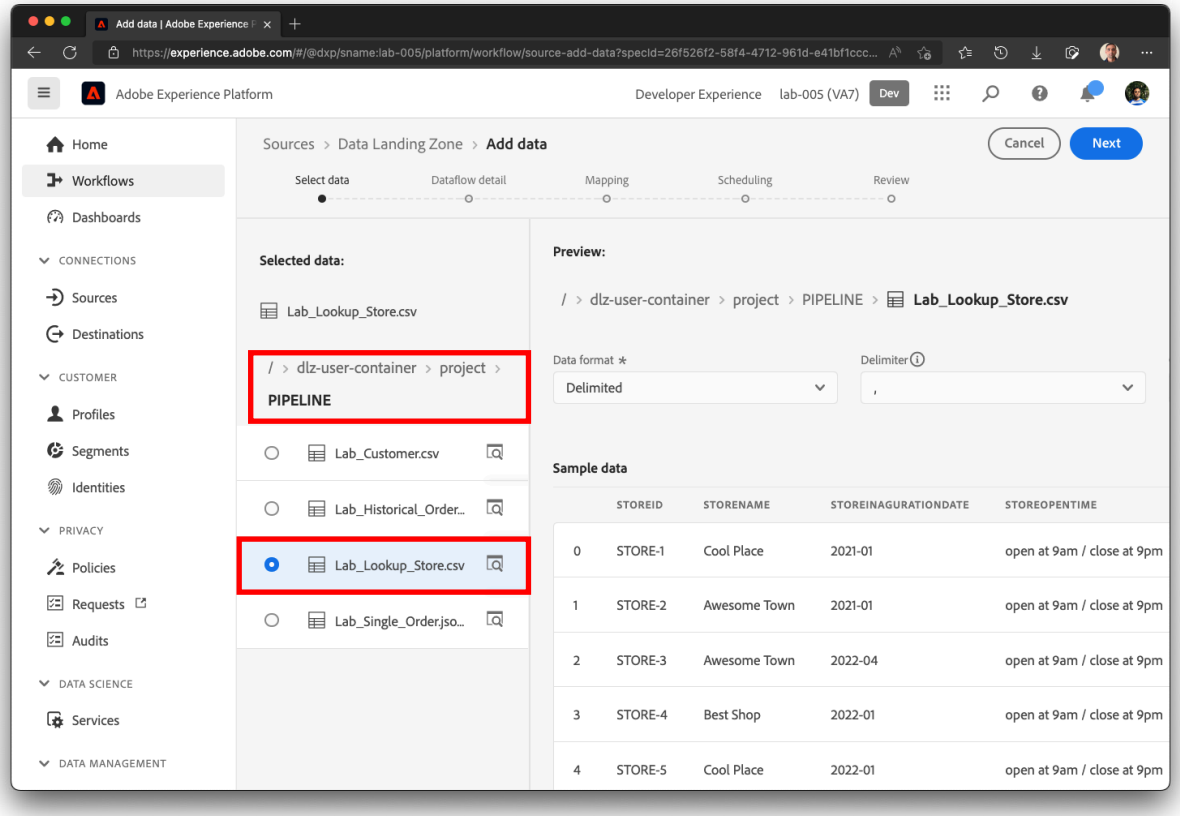
Go to Adobe Experience Platform → **Sources** → **Catalog**. In the **Cloud Storage** connectors, click **Setup** / **Add Data** in the **Data Landing Zone** card.

Tip	If at least one connection exists for that source, you will see "**Add data**" as the default action. If no connections exist for that source, you will see "**Setup**" as the default action.



In the next screen, navigate to **project** → **PIPELINE** and select the **Lab_Lookup_Store.csv**. On the right hand side, Data format should already be set to **Delimited**. Preview should load automatically. Click **Next**.

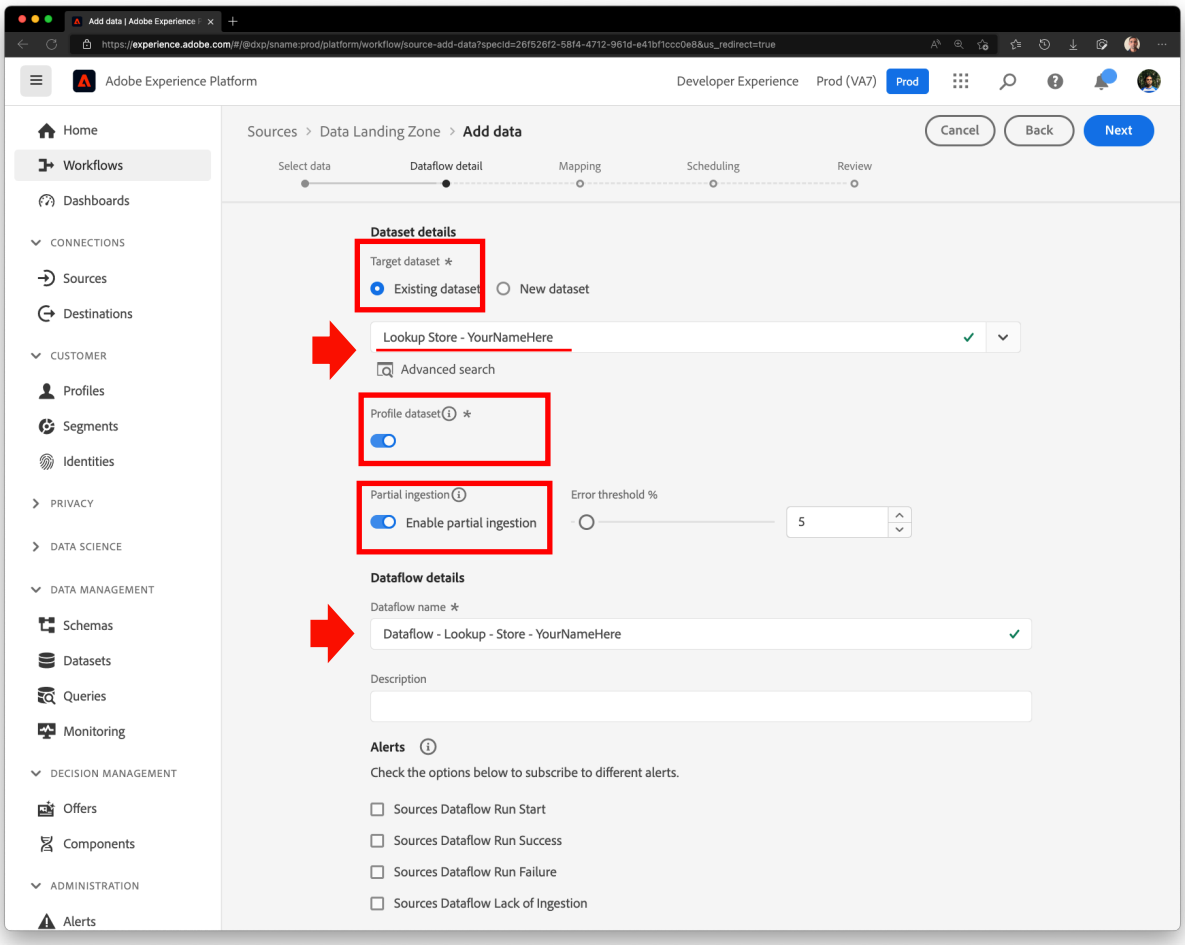
Tip	Check the list of delimiters and compression types supported by clicking on the corresponding drop downs



3.2.Define the target dataset

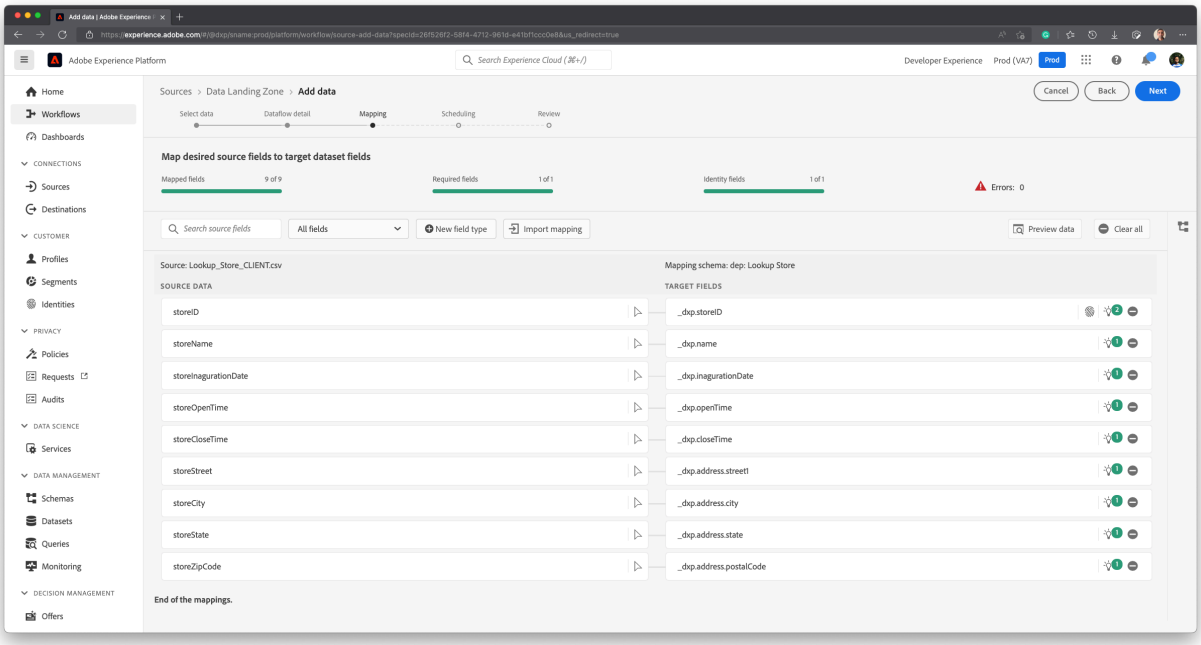
In the **Dataflow detail** screen, choose the **Existing dataset** Lookup Store - YourNameHere. **Error Diagnostics** is already turned ON. Now, turn ON **Enable Partial Ingestion**. Set the Data flow name as Dataflow – Lookup – Store - YourNameHere and click **Next**.

Tip	When Partial Ingestion is enabled, Error diagnostics are automatically enabled and hence the toggle box will disappear.



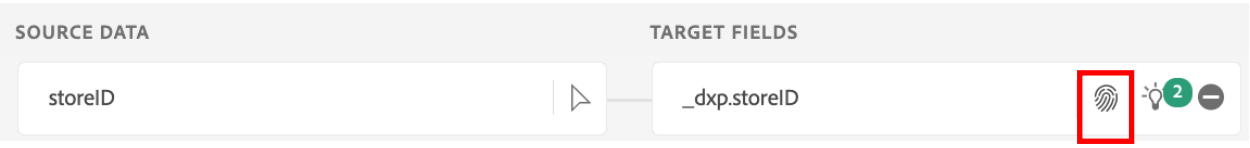
3.3.Data Prep / Transformation

Data Prep (Mapping) step will load. The ML Recommendation Service will pre-populate the mappings for you. Please review and ensure the mappings are as follows. If all mappings look good, click **Next**.

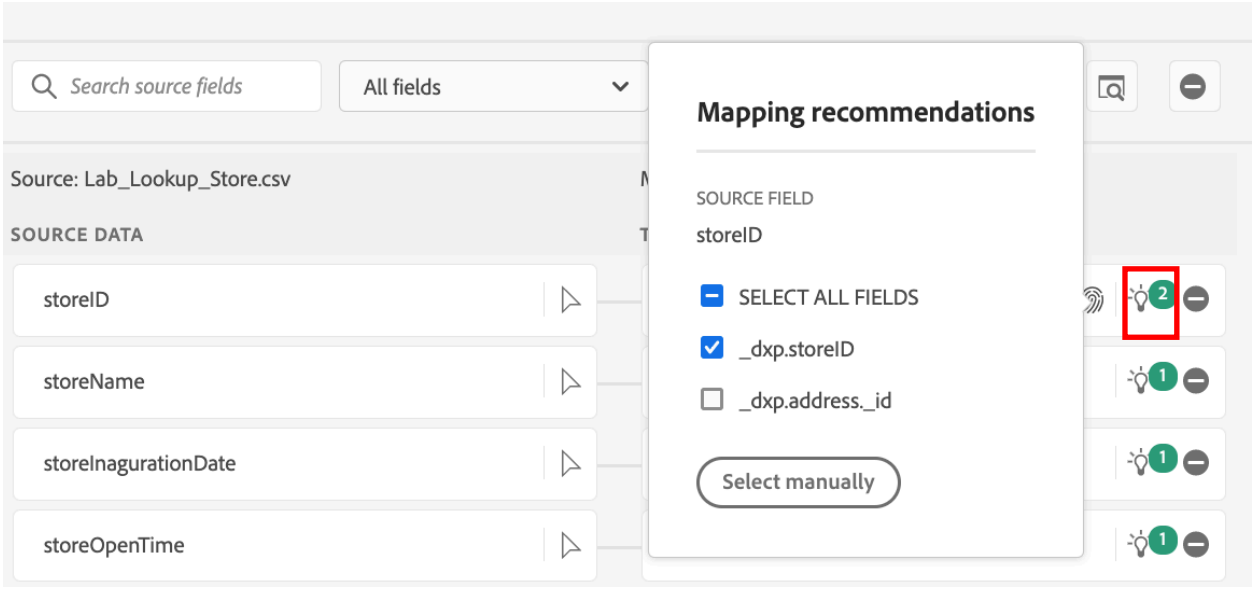


Learnings

Identity attributes are represented with a fingerprint icon next to them



If the bulb icon next to the XDM attribute has a number greater than 1, it means there are other recommendations by the system that you can browse through and select.



Tip	If you would like to change the recommendation, always check the new recommendation before unchecking the existing recommendation. If you uncheck all recommendations (with none checked), system treats it as an indication from the user that it got all recommendations for that source attribute wrong.

Click **Next** to proceed to schedule screen.

Final Mapping Set

The final mapping set should look like this

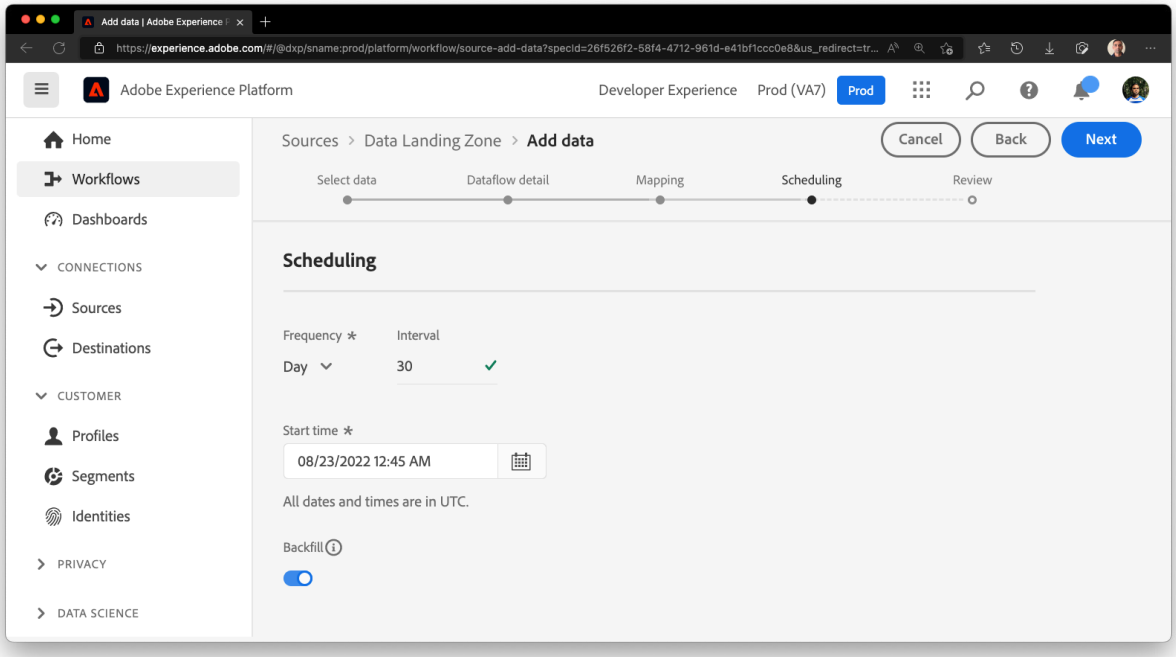
Source Column	XDM Property
storeID	_devbc.storeID
storeName	_devbc.name
storeInagurationDate	_devbc.inagurationDate
storeStreet	_devbc.address.street1
storeCity	_devbc.address.city
storeState	_devbc.address.state
storeZipCode	_devbc.address.postalCode
storeOpenTime	_devbc.openTime
storeCloseTime	_devbc.closeTime

Click **Next** to proceed to schedule screen.

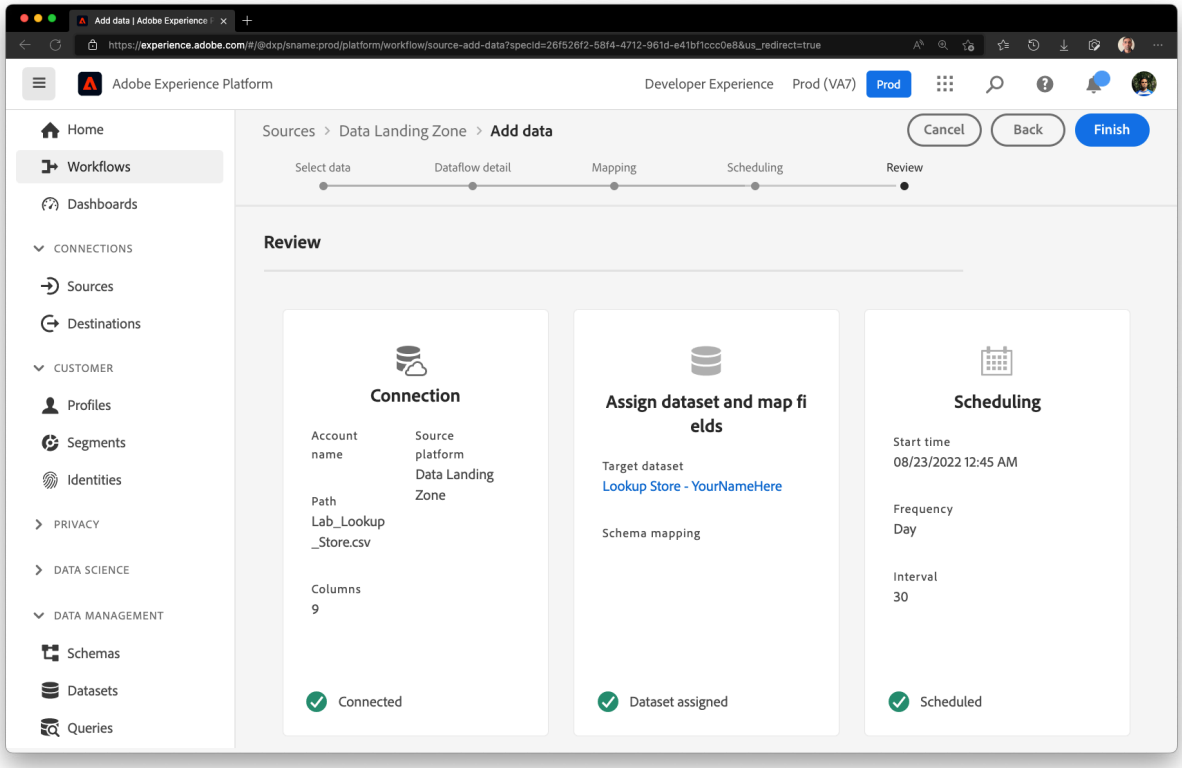
3.4.Schedule

The **Scheduling** screen will load. Change the **Frequency** to **Day** and set the **Interval** as **30**. Leave the **Backfill** toggle turned ON. This will schedule the Dataflow to run once every 30 days.

WARNING	Do not set the Frequency to Once. Currently One-time ingestion flows cannot be edited.
WARNING	Schedule times are in UTC. Not your local time. By default, the Start time is set to current time + 1 minute

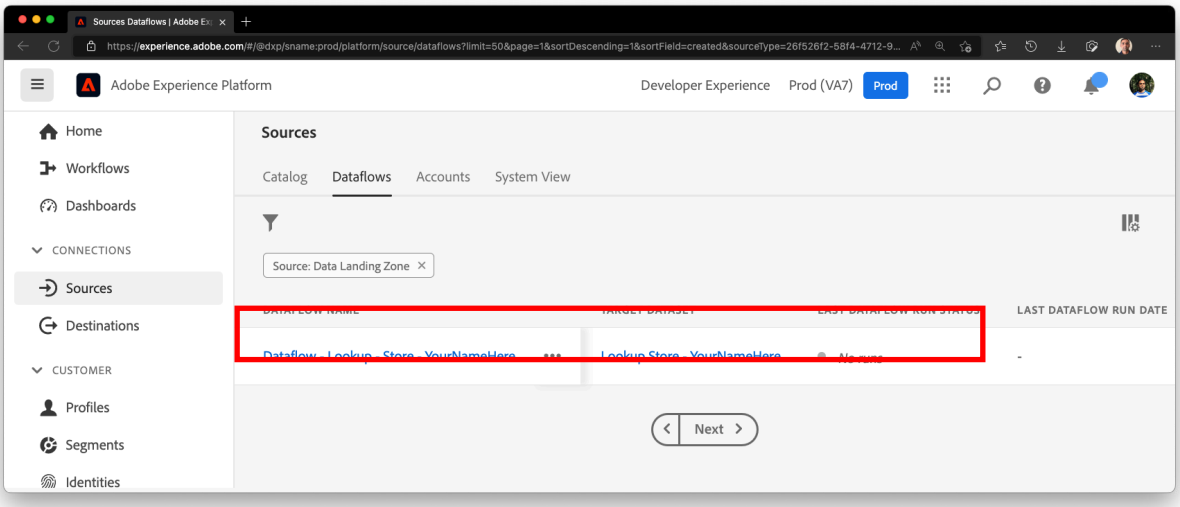


When the **Review** screen appears, click **Finish**.



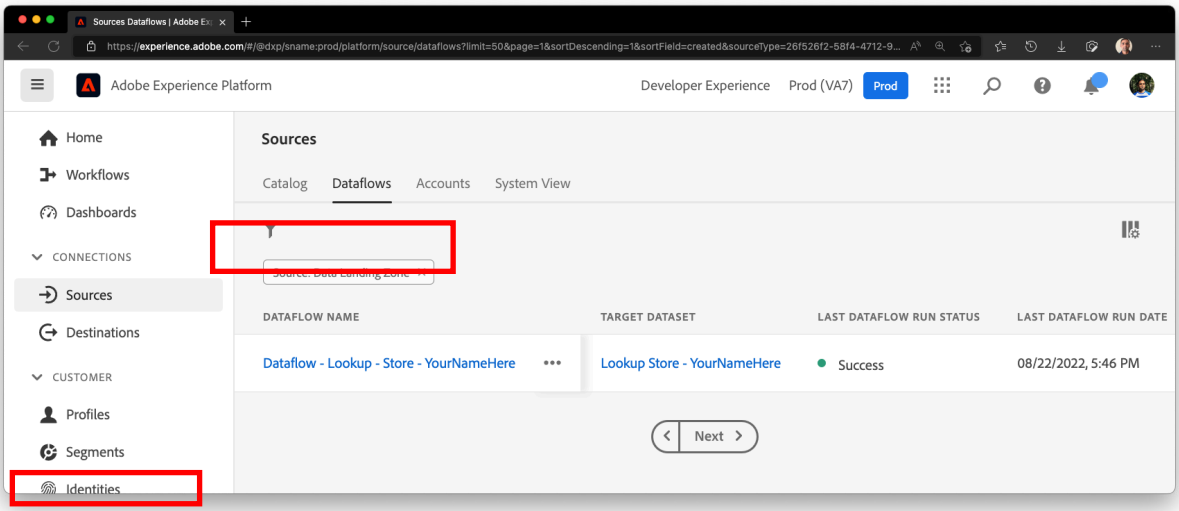
Lab Task – Monitoring the Dataflow

It takes few minutes to create the Dataflow. Once Dataflow is created, you will see the following screenshot. Notice that Last Dataflow Run Status indicates **No runs**. First run will kick off approximately in 5 minutes.

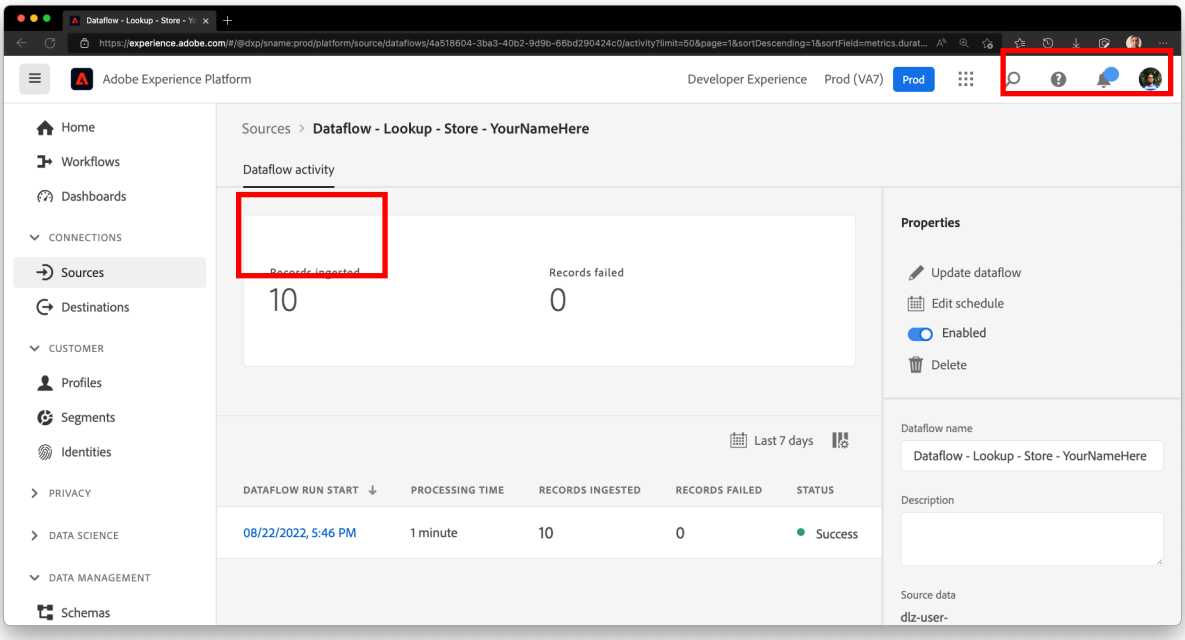


3.5. Check scheduled execution

After 5 minutes, refresh the page and the following will appear. Notice the **Last Dataflow Run Status** and **Last Dataflow Run Date**. Click on the Dataflow name that appears as blue link.



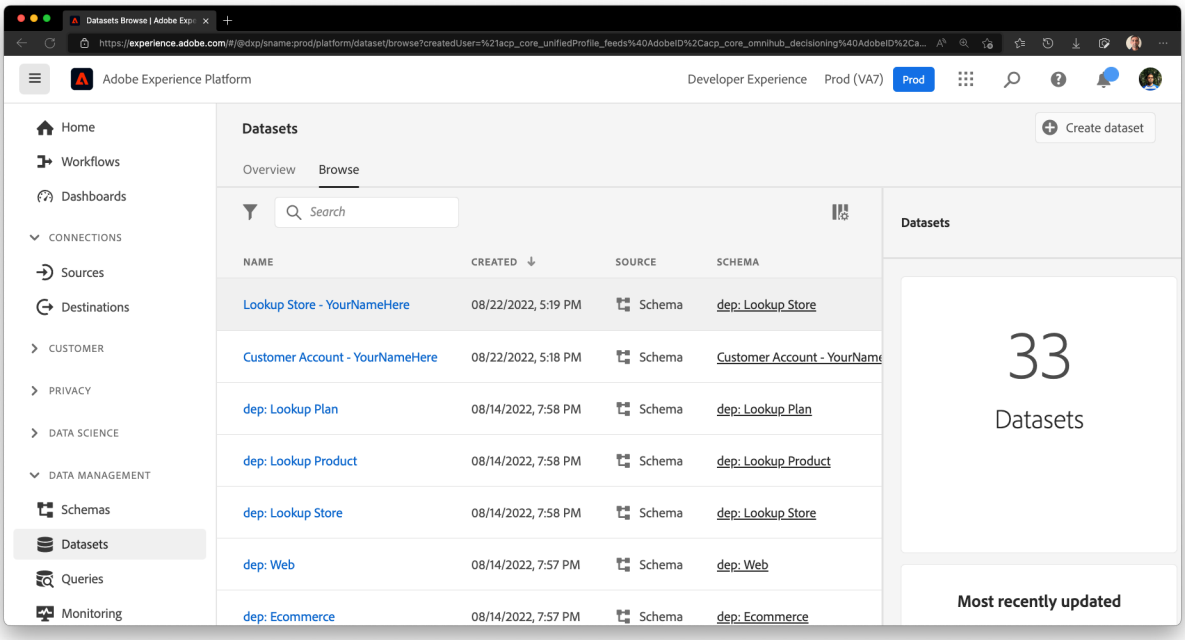
Click on the Dataflow name to get a list of Dataflow Runs. 10 Records should be ingested.



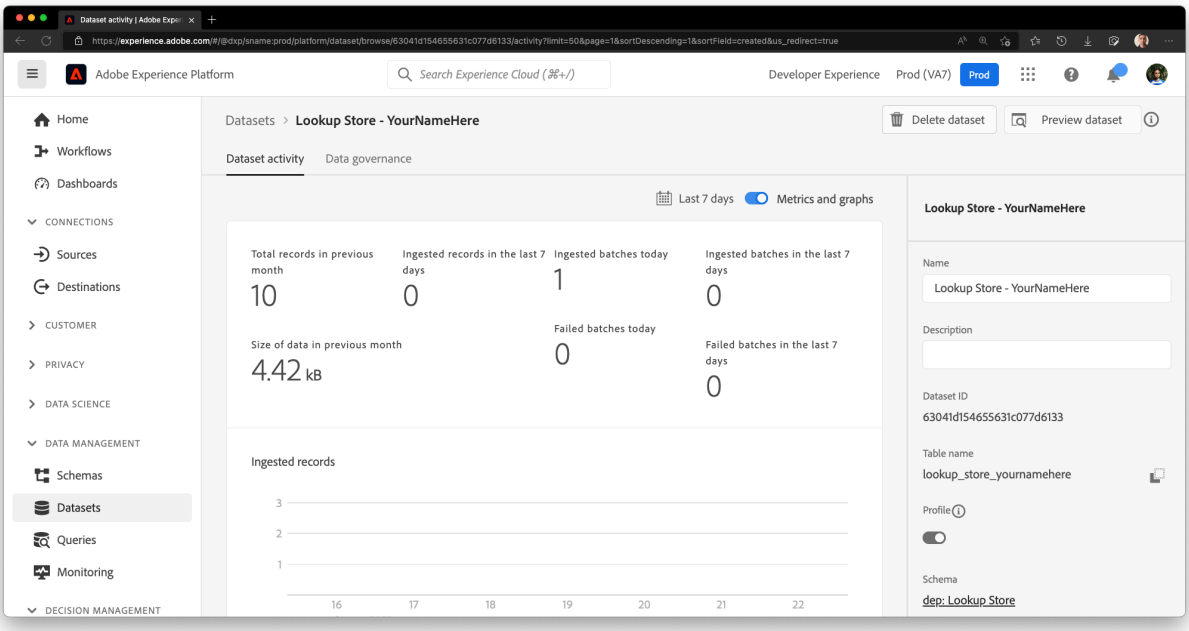
We will learn more about monitoring in the subsequent labs. For now, we will check that the data is loaded successfully.

Lab Task – Verify the data

In the Left Nav bar, Go to **Datasets** and on the right panel, click on **Lookup Store – YourNameHere** that appears as a blue link.



You will notice there are 10 records ingested. Click on the **Preview Dataset**



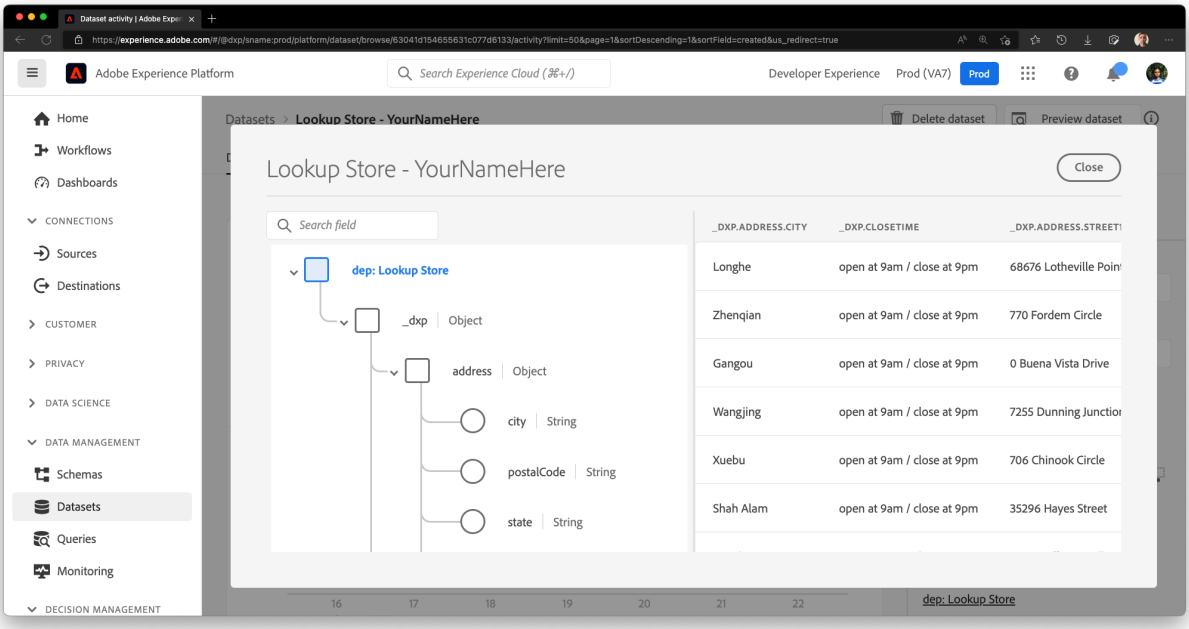
Data is loaded and displayed. Expand the XDM tree on the right to see child attributes.

TIP	When you click on a node such as **_dxp → address**, the right hand side will update to show only attributes within the selected parent node (**address** in this example). To see all attributes, click on the master node (**Lookup Store - YourNameHere** in this example).

Adobe Experience Platform
Lab 2 - Data Pipeline Customer Accounts

Adobe Experience Platform Bootcamp Deep Dive Edition

Name _____
Sandbox _____



1. Lab Overview

Ingest Customer Accounts data into AEP in batch mode using Data Landing Zone. This Lab will leverage delimited data on Landing Zone and ML Recommendations to map most of the source data. This job will be scheduled to run every 7 days.

This lab will introduce you to customizing the ML recommendations in Data Prep and updating manual mappings.

Expected time: 30 minutes

2. Learning Objectives

What should you walk away with after taking this Lab?

- Adding pass through mappings
- Using preview to check any data quality issues
- Address basic data quality issues using Calculated Fields

3. Lab Tasks – Ingest Customer Account data

In this exercise, we will load the Customer Account data from Data Landing Zone to AEP Data Lake and Profile.

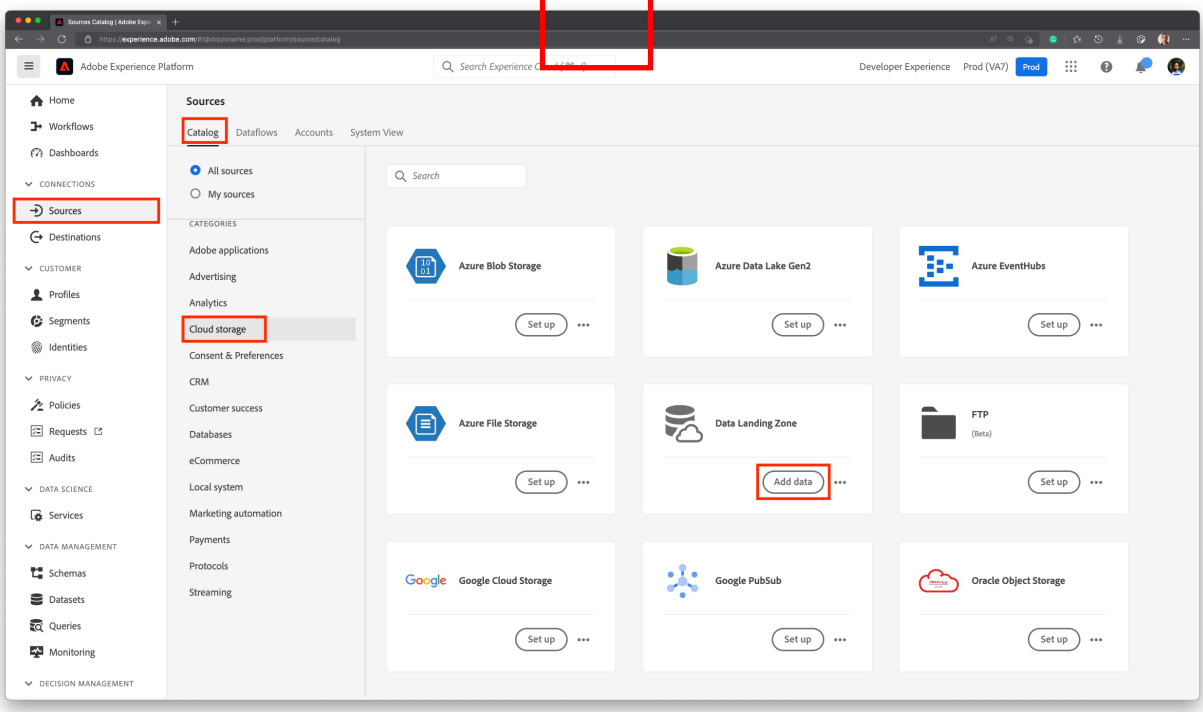
Pre-requisites

1. Customer Account JSON file uploaded in the Azure ADLS Directory

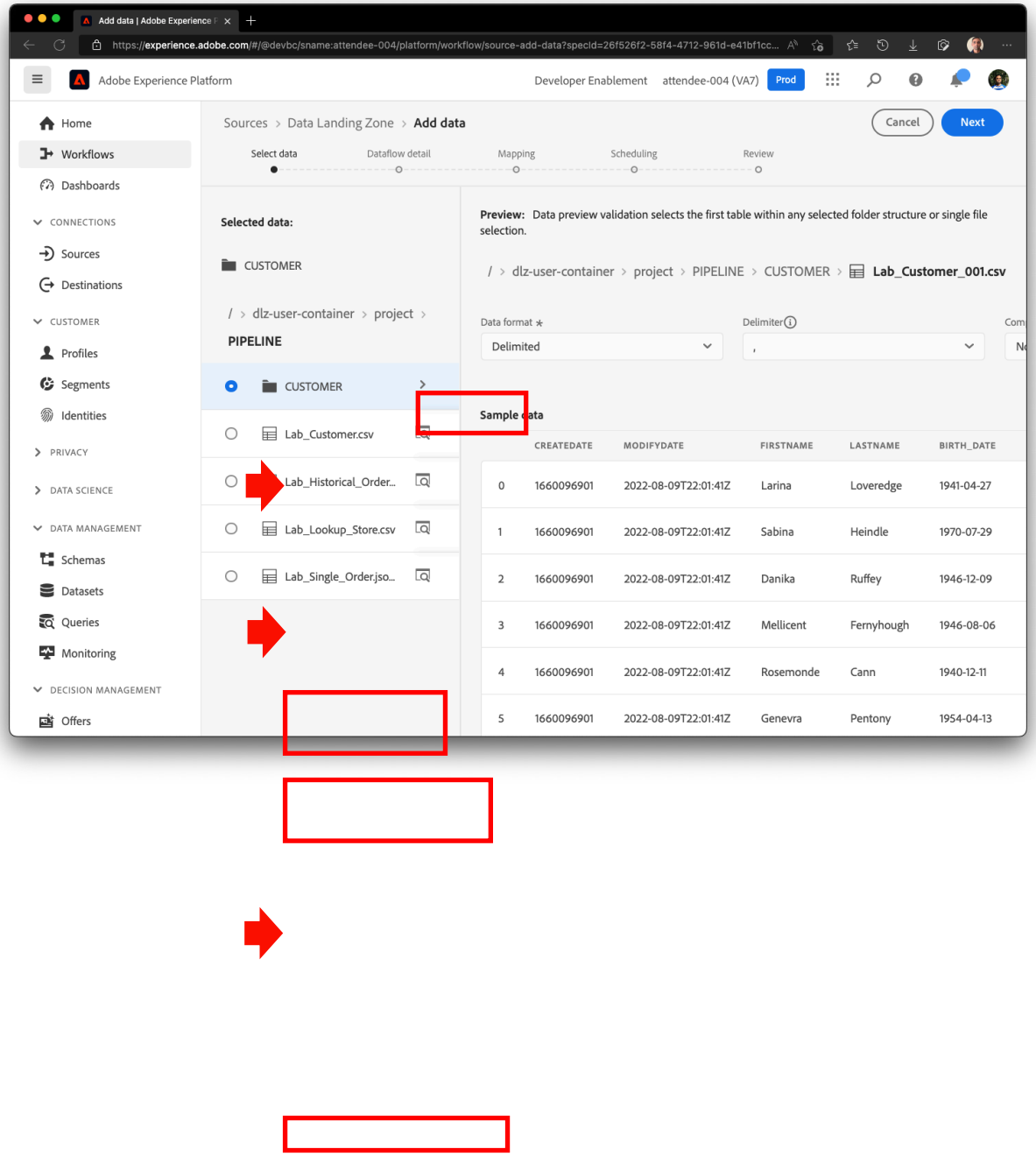
3.1. Select Source data

Go to Adobe Experience Platform à **Sources** à **Catalog** à **Cloud storage**. Click on **Setup** / **Add Data** for the Data Landing Zone.

Tip	If at least one connection exists for that source, you will see "**Add data**" as the default action. If no connections exist for that source, you will see "**Setup**" as the default action.

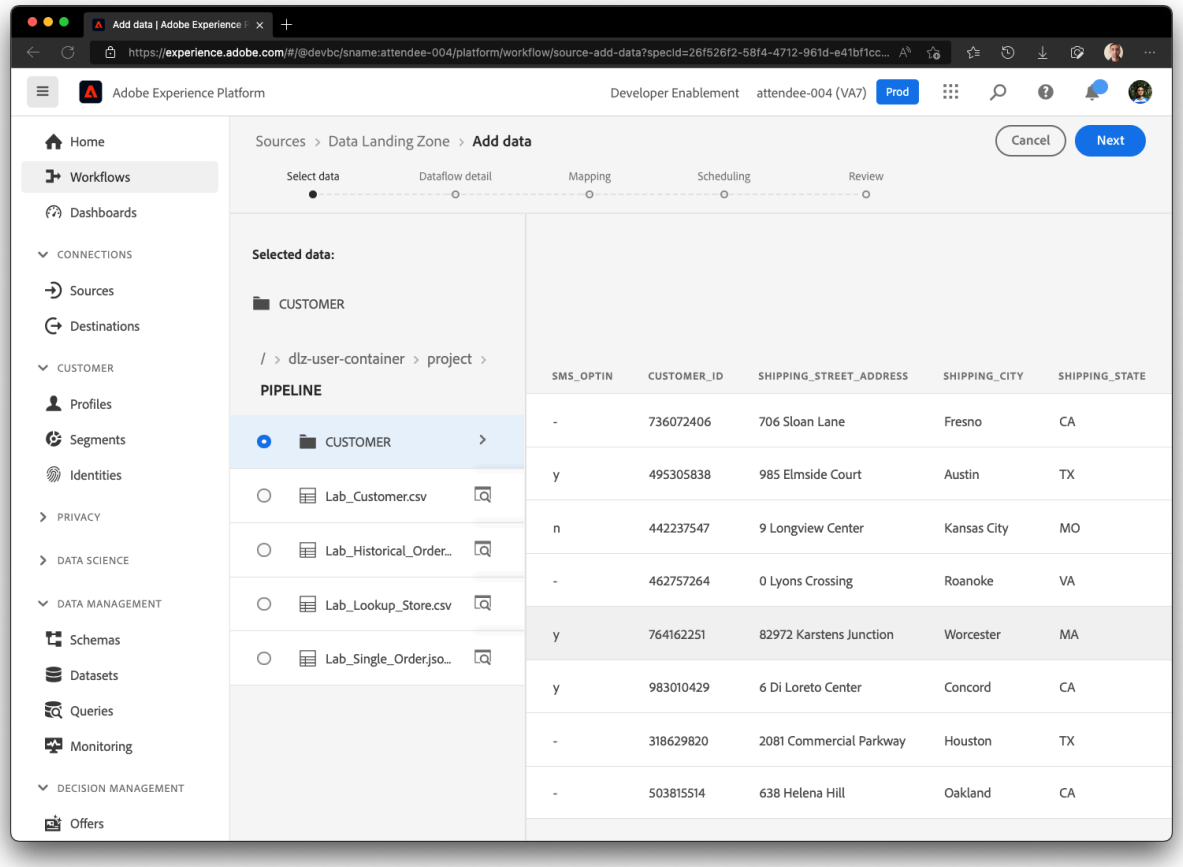


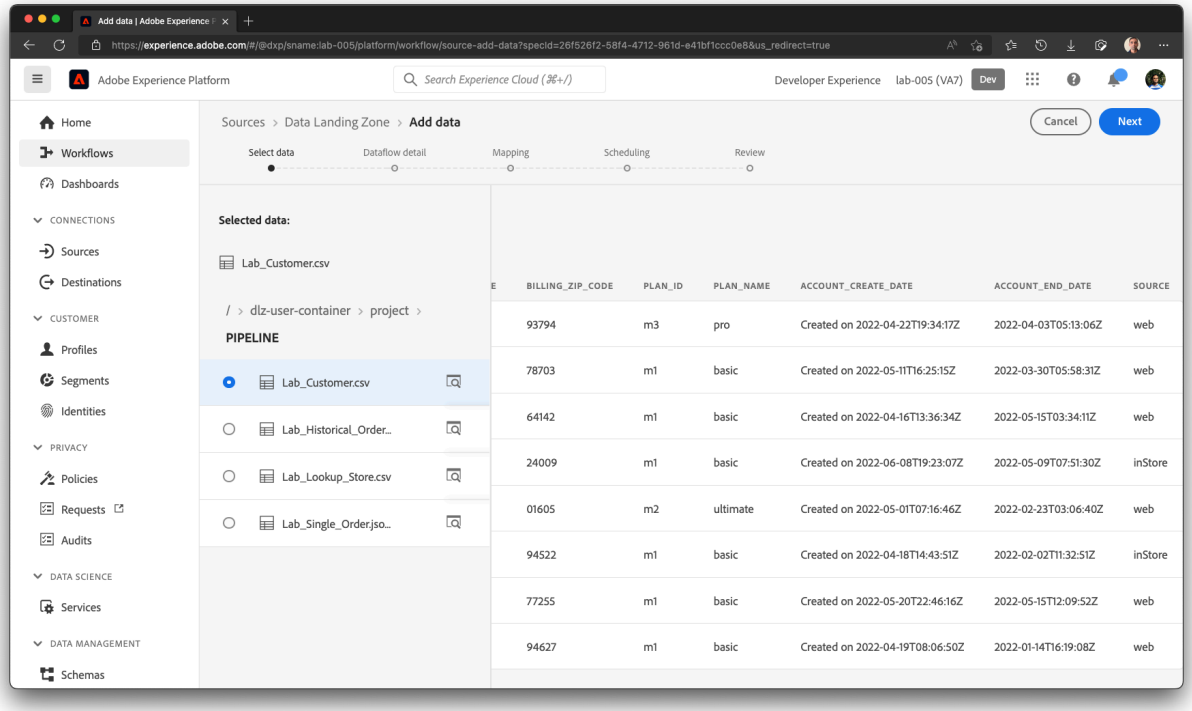
In the **Select data** screen, navigate to **/dlz-user-container/project/PIPELINE.** Select the radio button next to **CUSTOMER** folder. Do NOT click on the folder name. On the righthand side, **Delimited** is automatically selected as Data format. Notice on the right side that the first file in the folder is automatically selected **Lab_Customer_001.csv**.



In the preview here, notice the following attributes:

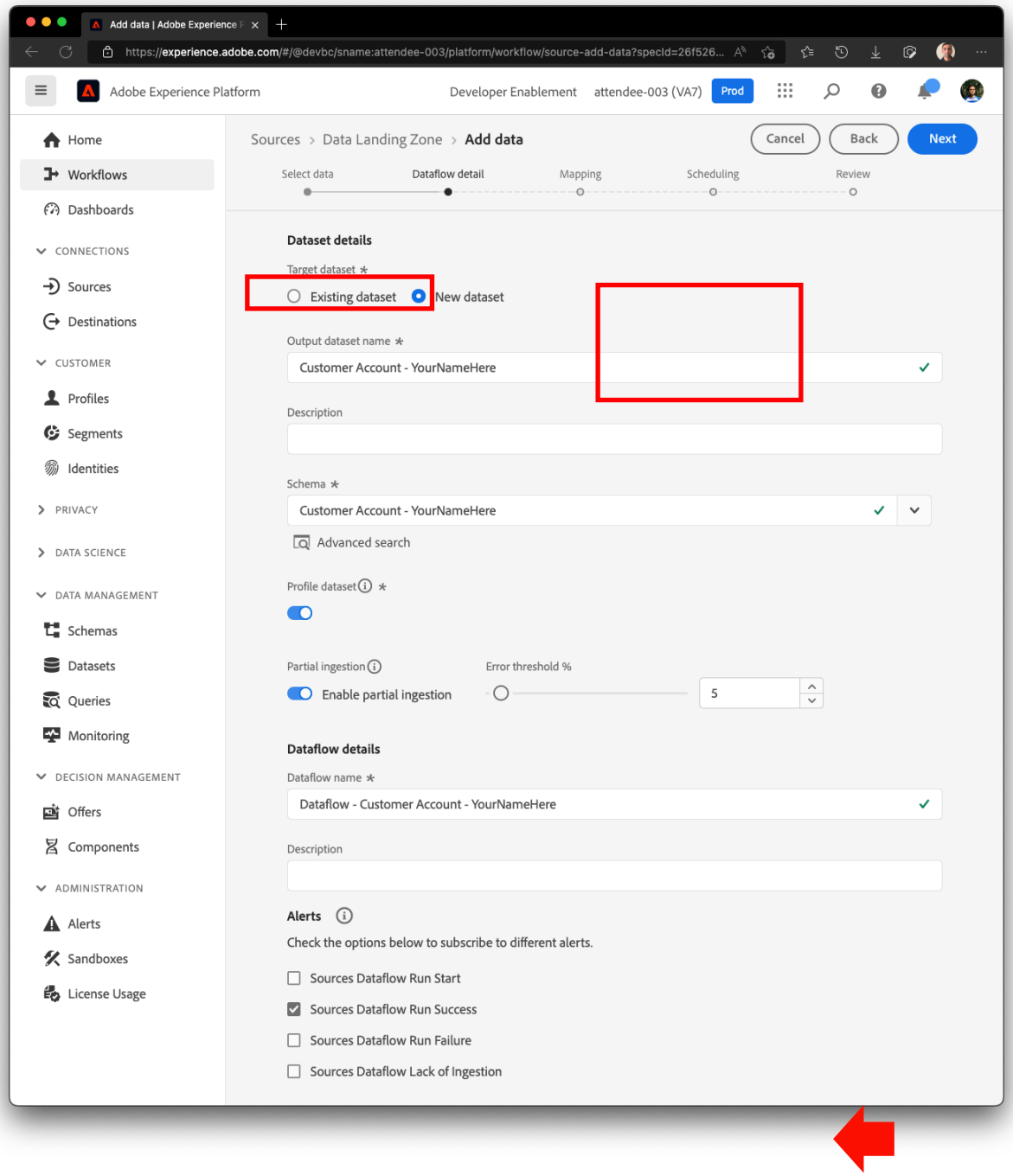
- **sms_optIn** has null values (shown in preview as -)
- **account_create_date** has string values along with date and time values. **account_end_date** has proper date format.





3.2.Define the Target Dataset

In the **Dataflow detail** screen, choose **New dataset**. Name the output dataset as **Customer Account - YourNameHere**. Select the schema name **Customer Account – YourNameHere**. Turn ON the **Profile dataset** toggle box. Turn ON the **Enable partial ingestion**. Set the Dataflow name as **Dataflow - Customer Account – YourNameHere**. Turn ON the alert **Sources Dataflow Run Success**. Click **Next** to continue.

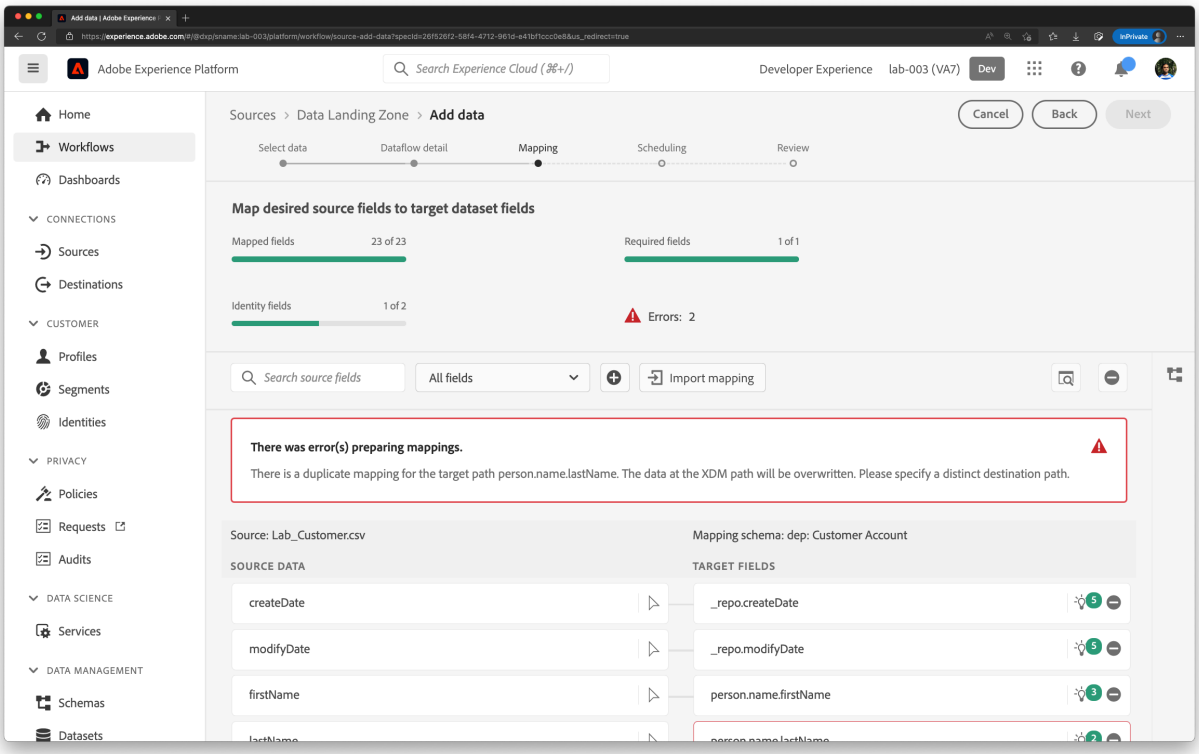


3.3.Data Prep / Transformation

In the **Mapping** screen, Recommendation engine will map several attributes automatically. However, there will be some errors that need to be addressed before moving on to next step.

The initial screen will look like this

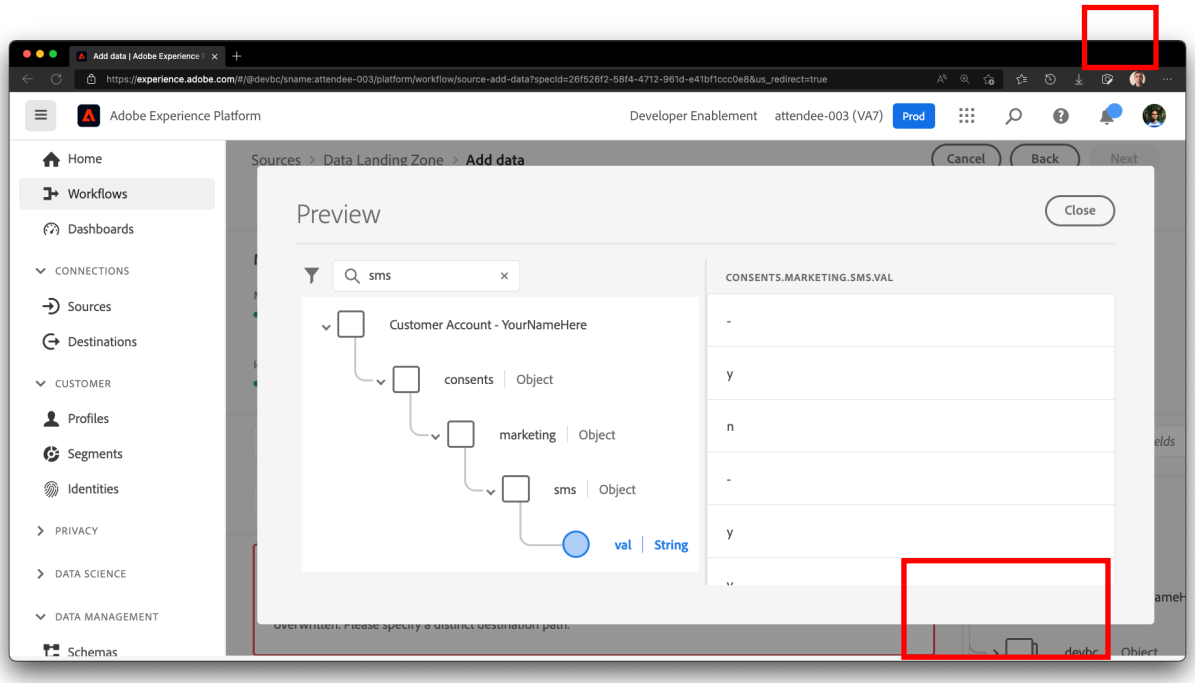
Warning	Due to the ML Recommendations, your screen may look different than the screenshot below.

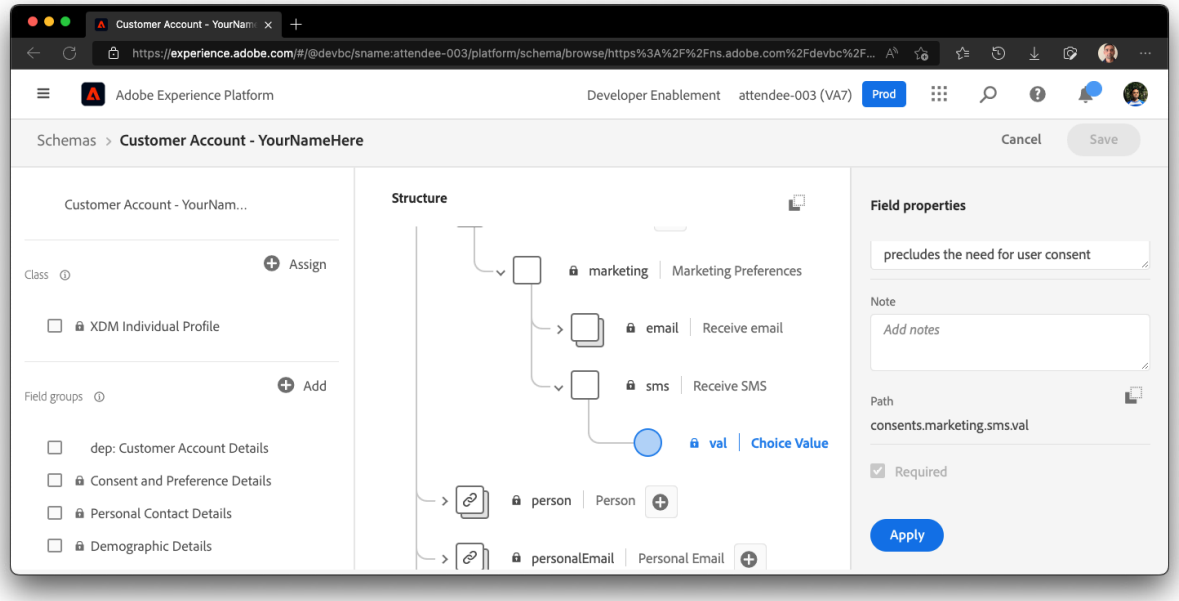


3.4.Preview the mapping output

SMS Opt In

Preview the mapping output by clicking  In Preview screen, navigate to the consents. marketing.sms.val attribute and you will notice that the mapping output has NULL values shown as hyphens (-). This is consistent with what we saw in the Preview of source in source object selection screen. However, this attribute is marked as a required attribute as seen in the schema screenshot below.



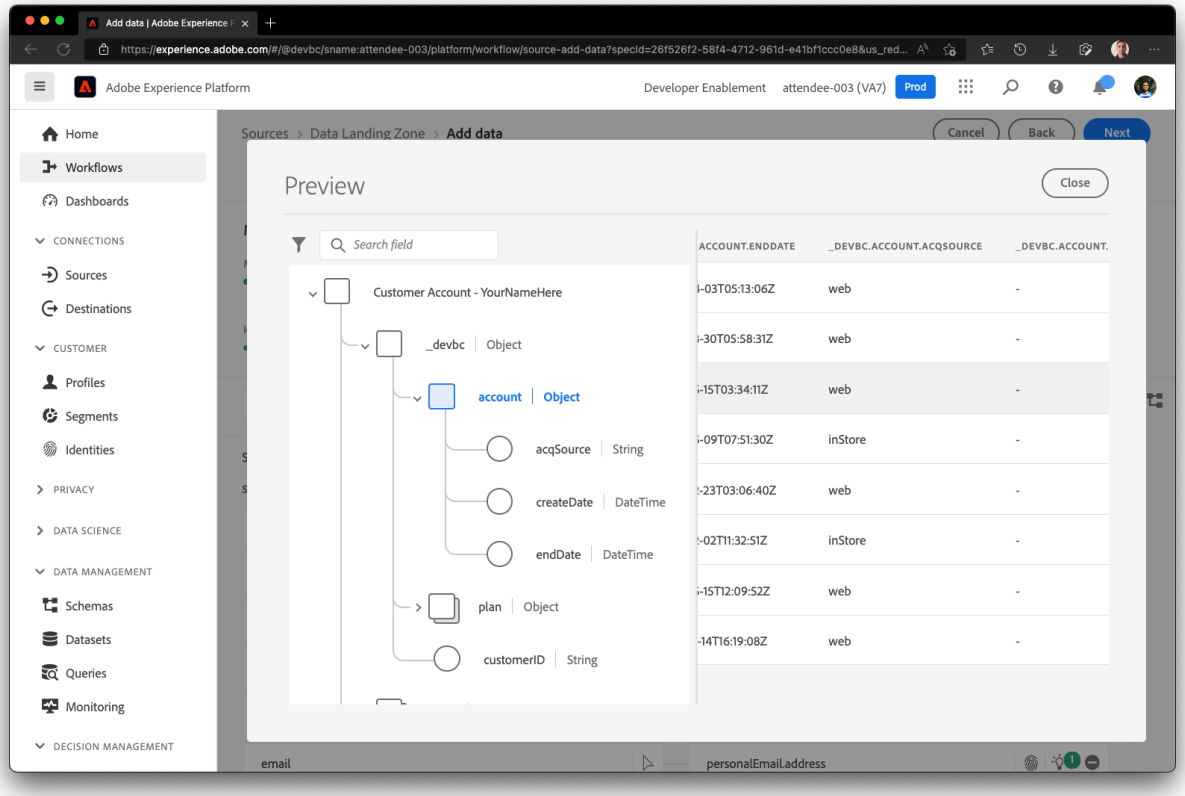


Learnings

At runtime, the rows without a SMS value will be rejected. If Partial ingestion is not enabled, this will fail the entire batch. If partial ingestion is enabled, the rows will NULL value will be rejected and other rows will be ingested

Account Create Date

In Preview screen, navigate to the **_devbc.account.createDate** .



Learnings

It is showing as NULL even though the source has value. This is because the source data contains string values such as "**Created on 2022-06-04T19:2409Z**". This is mapped to **_devbc.account.createDate** which is a timestamp data type. Since the input cannot be automatically converted into a timestamp, Data Prep now raises a warning (seen in Monitoring console when Dataflow runs) and NULLifies the result.

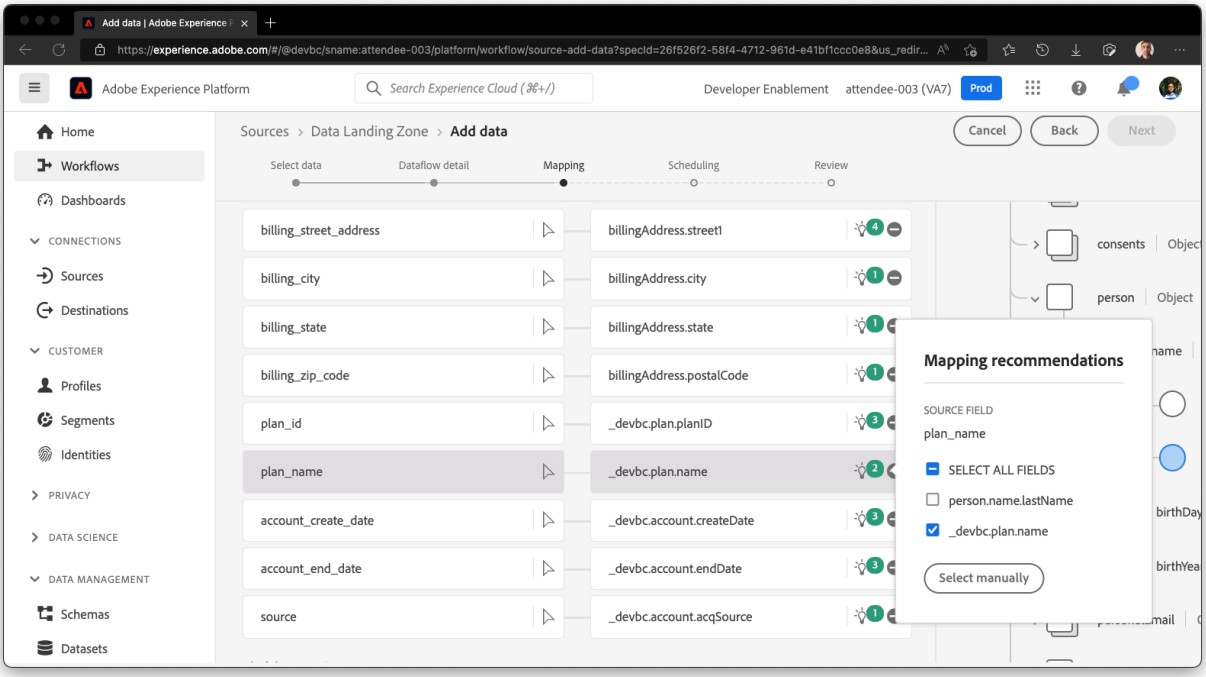
We will NOT fix this error as we will use this to understand how Data Prep / Mapper errors are reported later.

TIP	Note that the input file is CSV and all attributes are Strings. Data Prep automatically recognized the date attributes such as **account_end_date** and converted them into date. Similarly, Zip Code is automatically converted into number.

3.5. Re-map the attributes

When you scroll down, you will notice **plan_name** mapped to **person.name.lastName** (or some other invalid attribute). Click on the bulb icon (2) next to it and choose the **_devbc.plan.name**. The uncheck the **person.name.lastName**. If it is already mapped correctly, continue forward.

WARNING	Do not uncheck the **person.name.lastName** first. If you uncheck first, the ML recommendations will reset and you will have to select the **_devbc.plan.name** manually from the XDM schema on the right



Similarly, Change the mappings as necessary to match the table below. The table is sorted in Alphabetical order of XDM fields. The rows marked with★are incorrect and will be corrected in the next step.

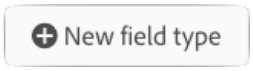
#	Source Column	XDM fields ↓	✓
1	source	_devbc.account.acqSource	
2	account_create_date	_devbc.account.createDate	
3	account_end_date	_devbc.account.endDate	
4	customer_id	_devbc.customerID	
5	plan_name	_devbc.plan.name	
6	plan_id	_devbc.plan.planID	
7	create_date	_repo.createDate	
8	modifyDate	_repo.modifyDate	
9	billing_city	billingAddress.city	
10	billing_zip_code	billingAddress.postalCode	
11	billing_state	billingAddress.state	
12	billing_street_address	billingAddress.street1	
13	email_optIn	consents.marketing.email.val	
14	sms_optIn	consents.marketing.sms.val	★
15	mobile_phone	mobilePhone.number	
16	birth_Date	person.birthDayAndMonth	★
17	first_name	person.name.firstName	
18	last_name	person.name.lastName	
19	email	personalEmail.address	
20	shipping_city	shippingAddress.city	
21	shipping_zip_code	shippingAddress.postalCode	
22	shipping_state	shippingAddress.state	
23	shipping_street_address	shippingAddress.street1	

3.6.Calculated Fields

We will now correct the two rows that were marked ★ by using Calculated fields to calculate the values.

SMS Opt In

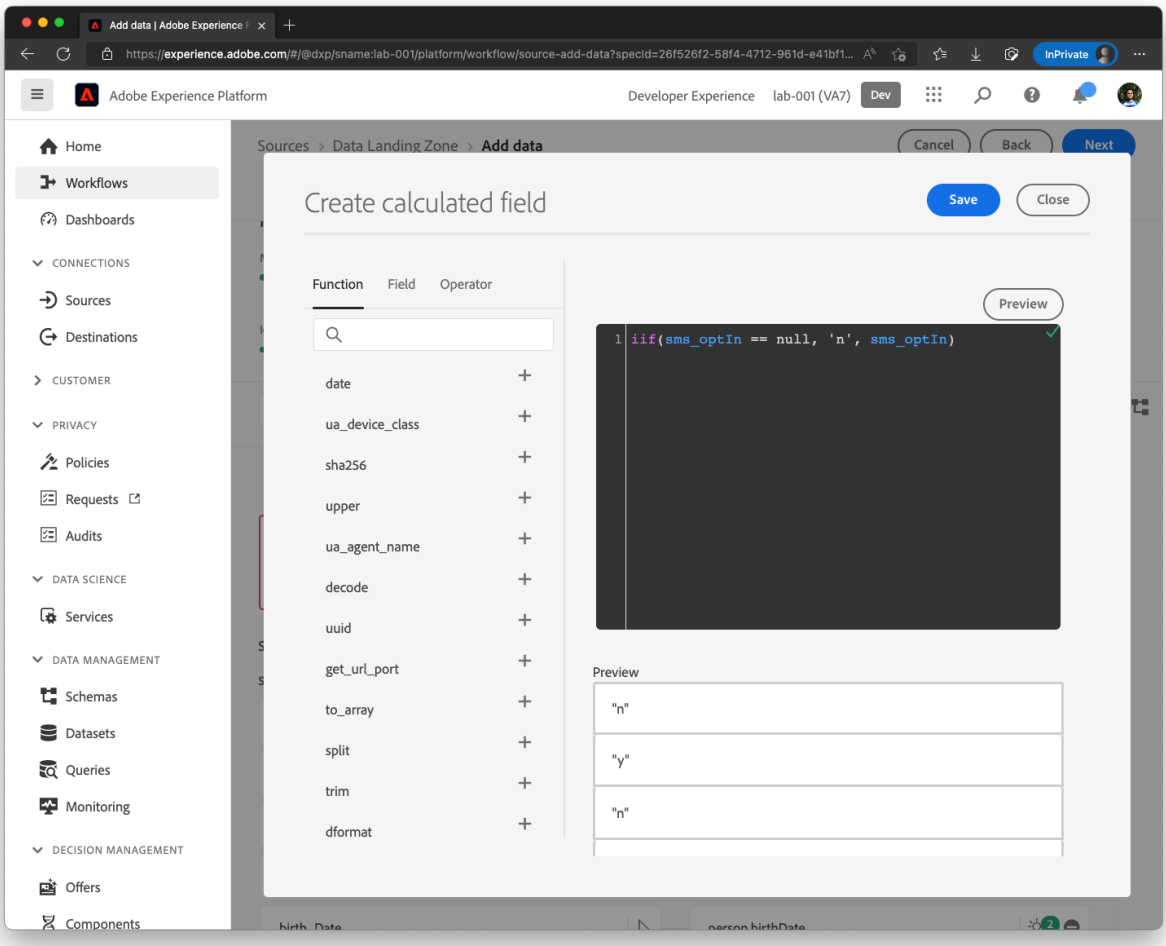
Scroll down to the mapping for XDM attribute **consents.marketing.sms.val**, and remove the mapping.

Now create a calculated field by clicking 

icon(sometimes shown as ) > **Add Calculated Field**. In the Create Calculated field dialog box write the following expression and hit **Preview**.

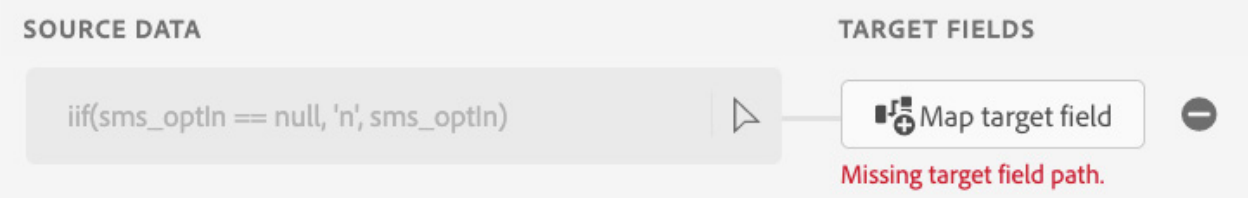
```
iif(sms_optIn == null, 'n', sms_optIn)
```

You should see a green checkmark in the top right corner of the black box indicating the validity of the expression and the data Preview should only show "n" or "y" as values. Then click **Save**.



NOTE	For the demonstration purposes of this lab, we are assuming lack of consent as an explicit No represented by "n". Different customers may have different way of handling lack of consent

The mapping will now look like this.



Click on Map target field and choose **consents.marketing.sms.val** attribute. Final mapping will look like this:

Learnings

Data Prep conditional operators can be used to cleanse input fields. Similar logic can be applied to email consent (**consents.marketing.email.val**)

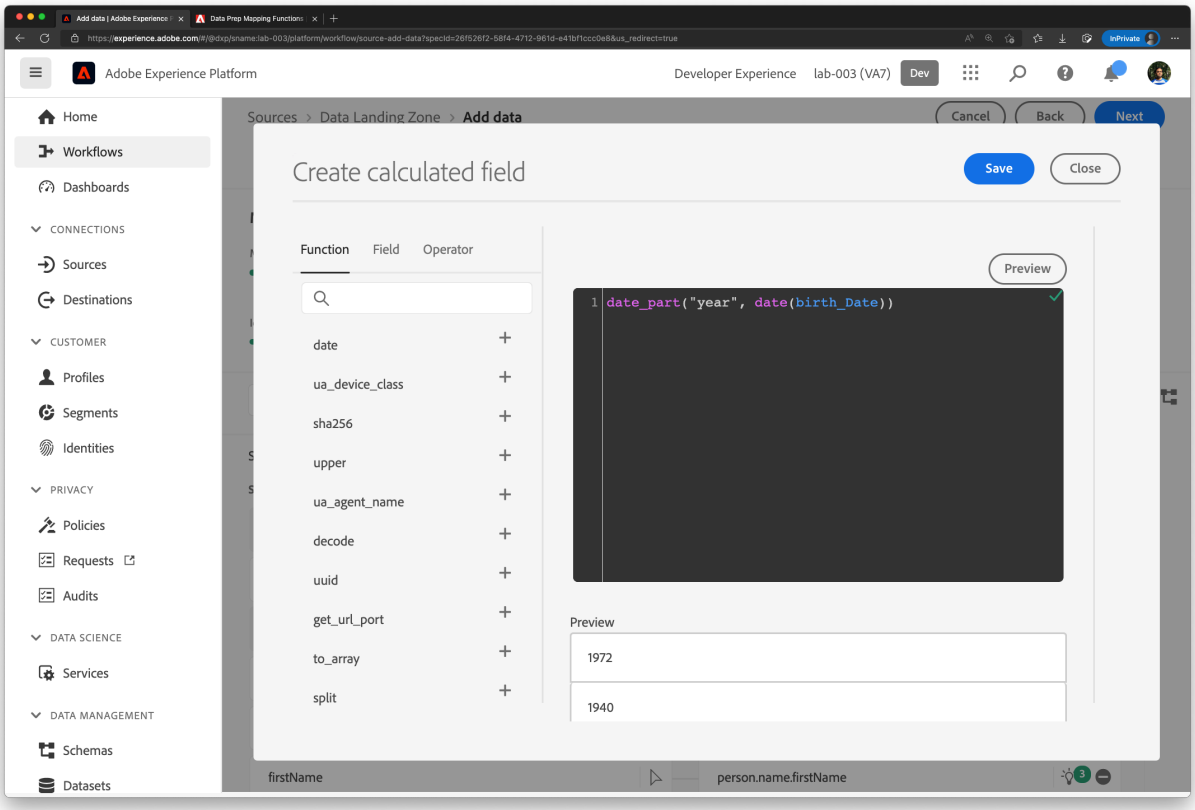
Birth Dates

Find the mapping associated with the XDM attribute **person.birthDayAndMonth** and remove that row by clicking on the (-) icon next to it.

If you find a mapping associated with **person.birthYear**, remove that row as well.

Add a Calculated Field by clicking the (+) icon and **Add Calculated Field**. Type in the following expression into the calculated field

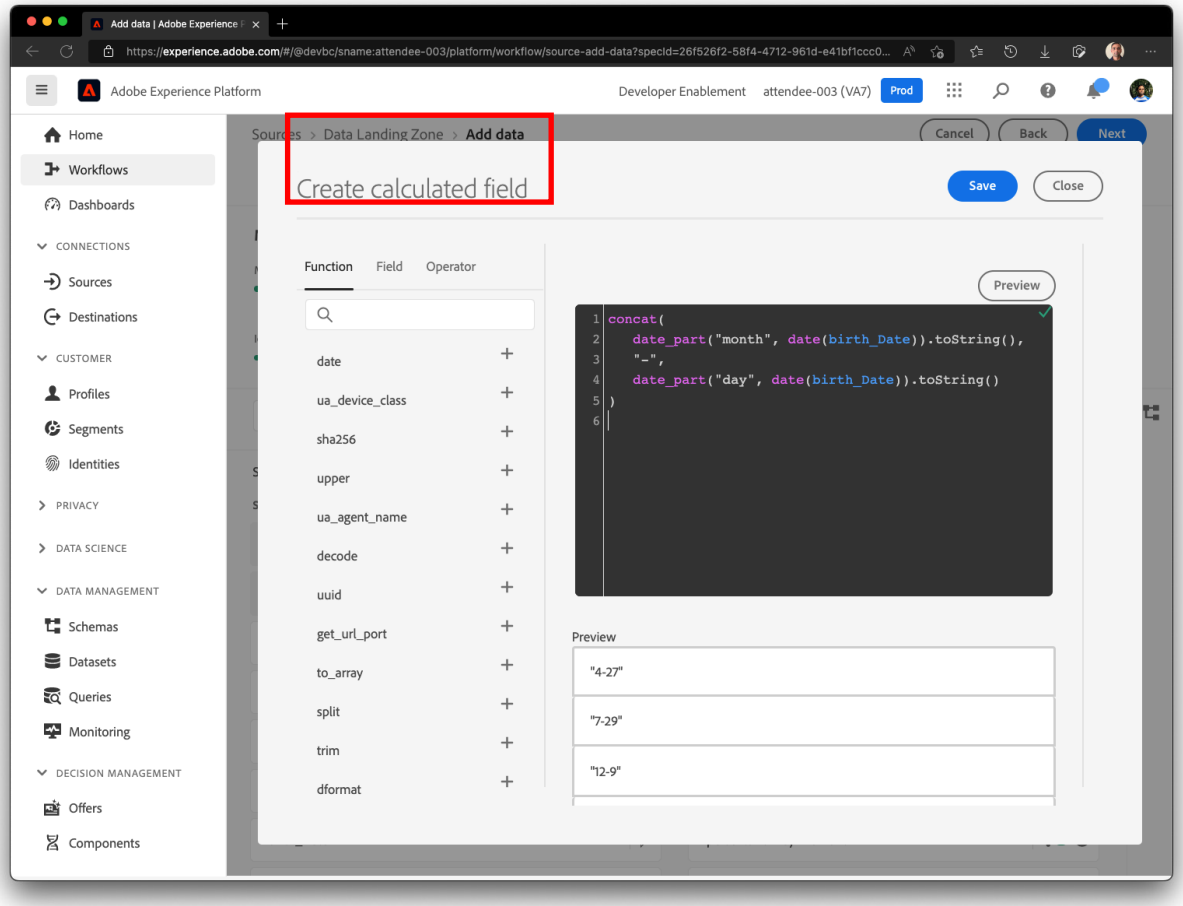
```
date_part("year", date(birth_Date))
```



Click Save to save this expression and map it to **person.birthYear**

Add another calculated field with the following expression and map it to **person.birthDayAndMonth**

```
concat(
  date_part("month", date(birth_Date)).toString(),
  "-",
  date_part("day", date(birth_Date)).toString()
)
```



NOTE	Notice that the month for some rows is a single digit. For example, 4-27 instead of 04-27. This will cause the data to be failed. We will use that to check errors in the monitoring console.

Now click on the **Map target field** button and assign it to **person.birthDayAndMonth**. The mapping will now look as below.

Final Mapping Set

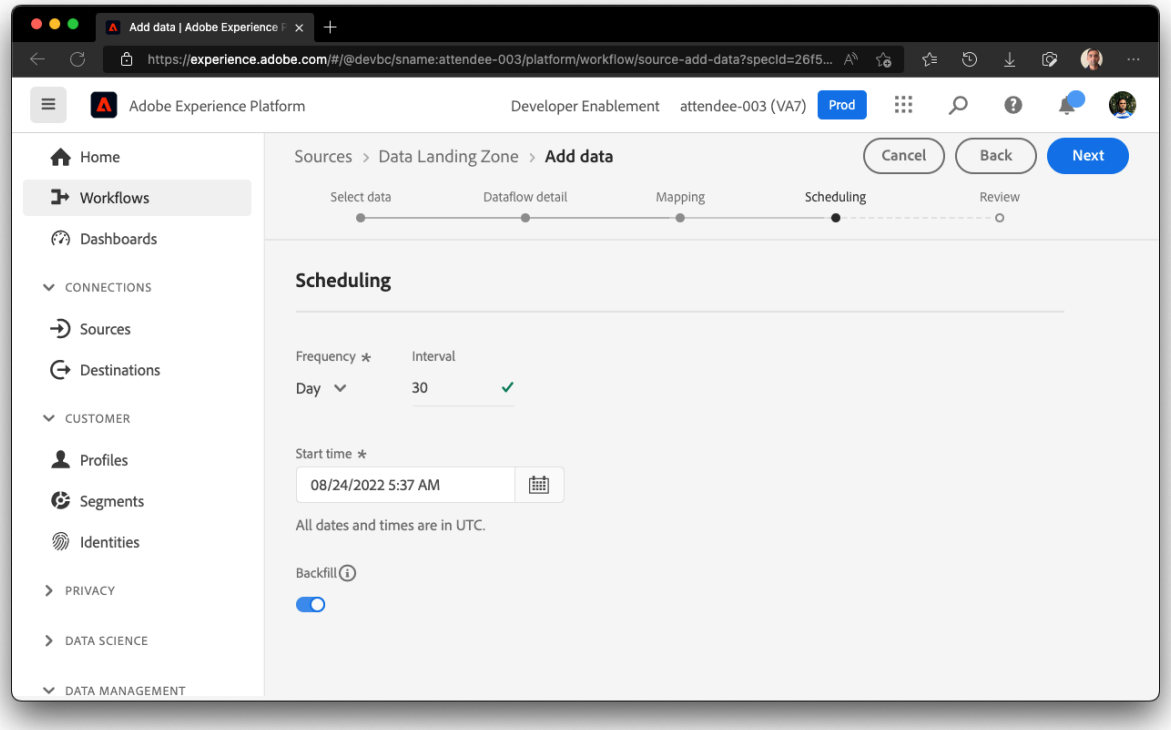
The final mapping set looks like below

#	SourceColumn	XDM field ↓	✓
1	Source	_devbc.account.acqSource	
2	account_create_date	_devbc.account.createDate	
3	account_end_date	_devbc.account.endDate	
4	customer_id	_devbc.customerID	
5	plan_name	_devbc.plan.name	
6	plan_id	_devbc.plan.planID	
7	billing_city	billingAddress.city	
8	billing_zip_code	billingAddress.postalCode	
9	billing_state	billingAddress.state	
10	billing_street_address	billingAddress.street1	
11	email_optIn	consents.marketing.email.val	
12	mobile_phone	mobilePhone.number	
13	first_name	person.name.firstName	
14	last_name	person.name.lastName	
15	email	personalEmail.address	
16	create_date	repo.createDate	
17	last_modified	repo.modifyDate	
18	shipping_city	shippingAddress.city	
19	shipping_zip_code	shippingAddress.postalCode	
20	shipping_state	shippingAddress.state	
21	shipping_street_address	shippingAddress.street1	

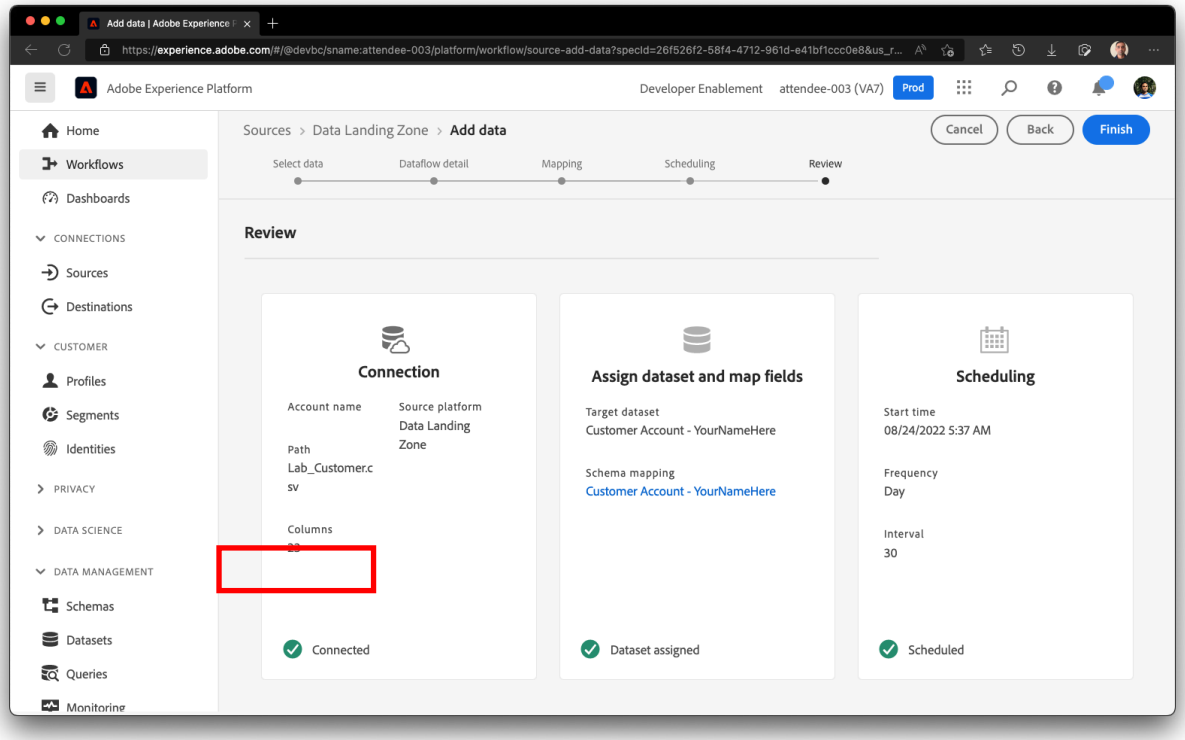
Calculated fields		✓
iif(sms_optIn == null, 'n', sms_optIn)	consents.marketing.sms.val	
concat(date_part("month", date(birth_Date)). toString(), "-", date_part("day", date(birth_Date)).to- String())	person.birthDayAndMonth	
date_part("year", date(birth_Date))	person.birthYear	

3.7. Schedule

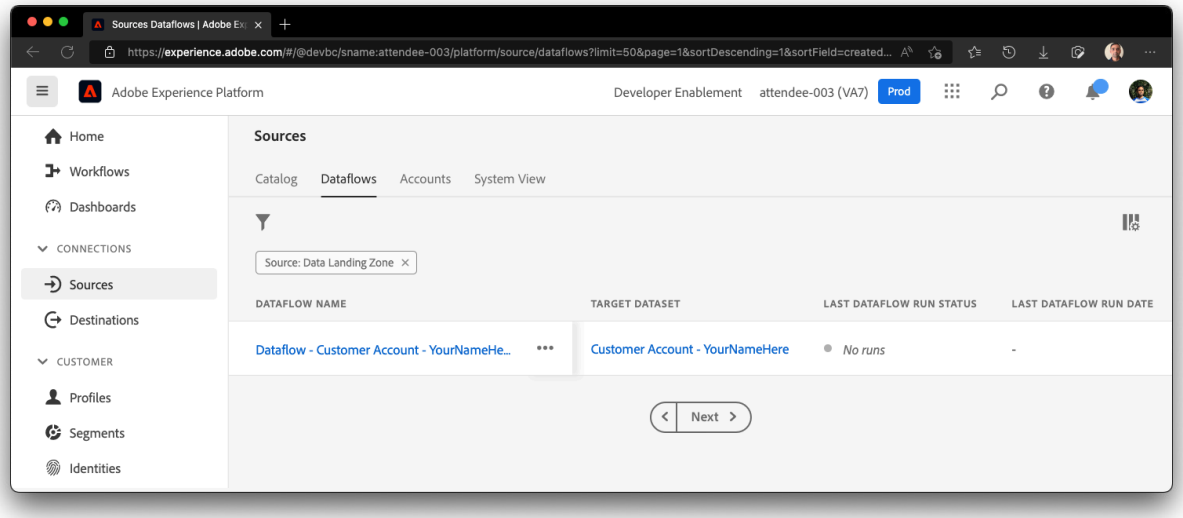
In the **Scheduling** step, set the **Frequency** to **Day** and **Interval** to **30** and leave the **Backfill** turned on. This will schedule the Dataflow to run every 30 days



Review the data flow and click **Finish**.

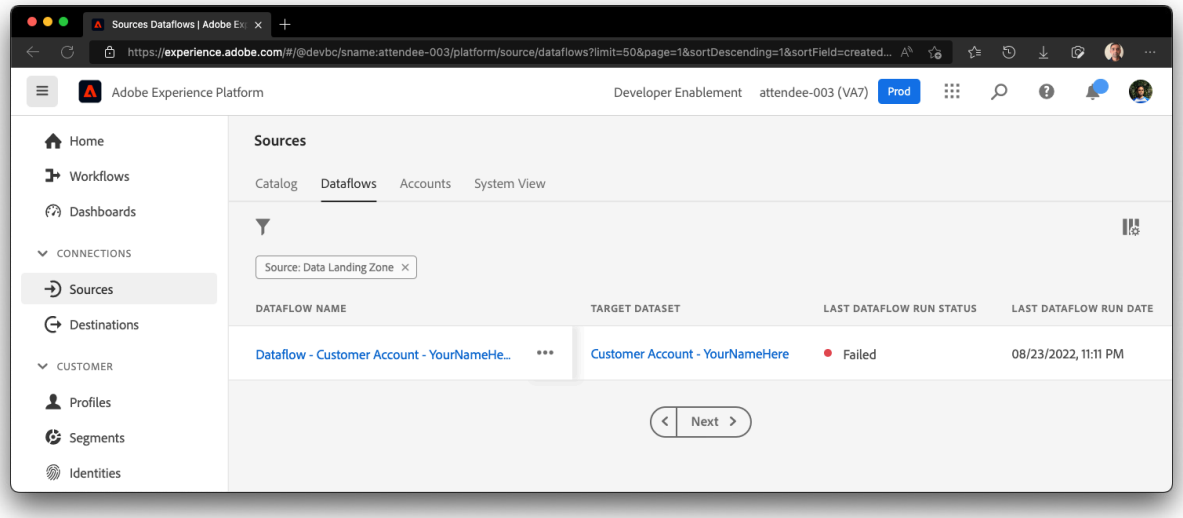


It takes few minutes to create the Dataflow. Once Dataflow is created, you will see the following screenshot. Notice that Last Dataflow Run Status indicates **No runs**. First run will kick off approximately in 15 minutes.

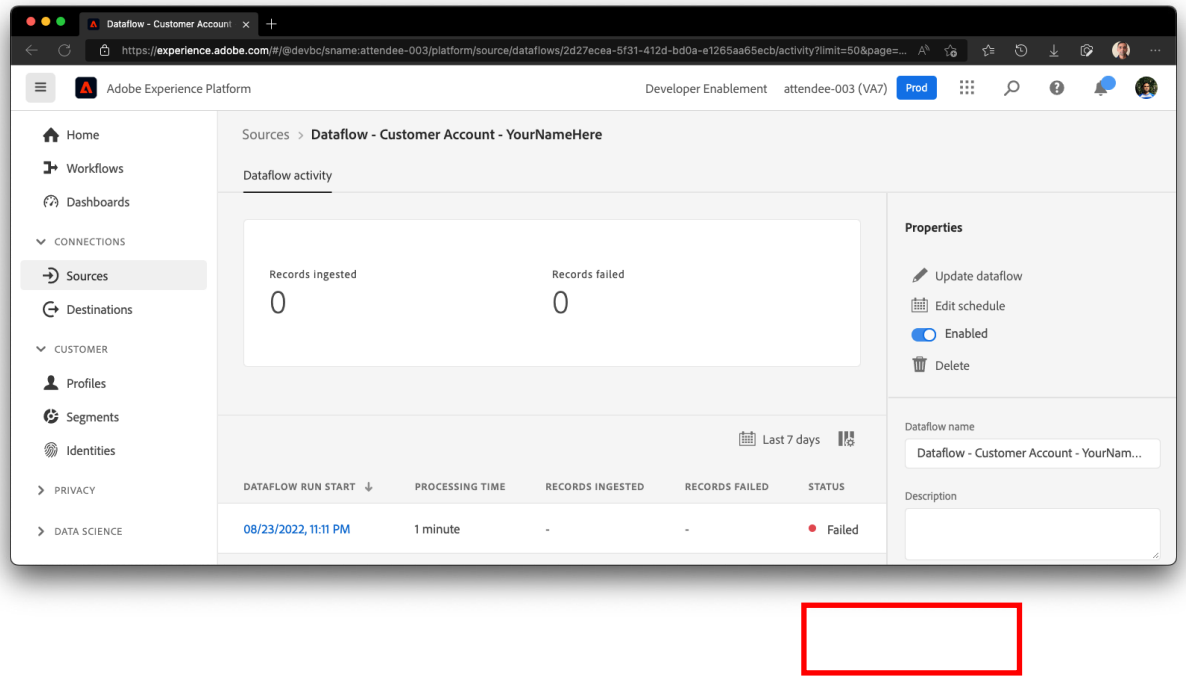


3.8.Check execution status and error messages

After 15 minutes, the following will appear. Notice the **Last Dataflow Run Status** and **Last Dataflow Run Date**. Click on the Dataflow name that appears as a blue link

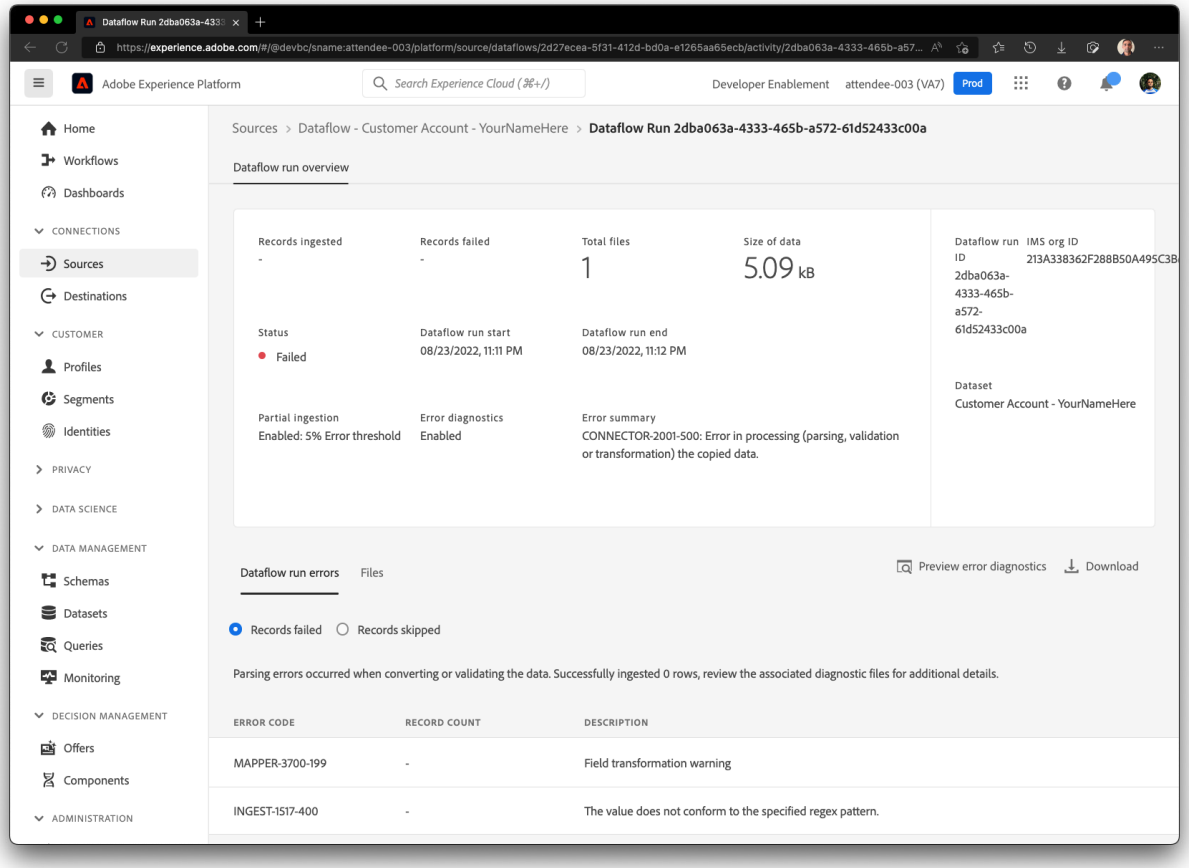


We will notice that there is one Dataflow run and in Failed status. Click on the Dataflow run with timestamp that is listed as a blue link.

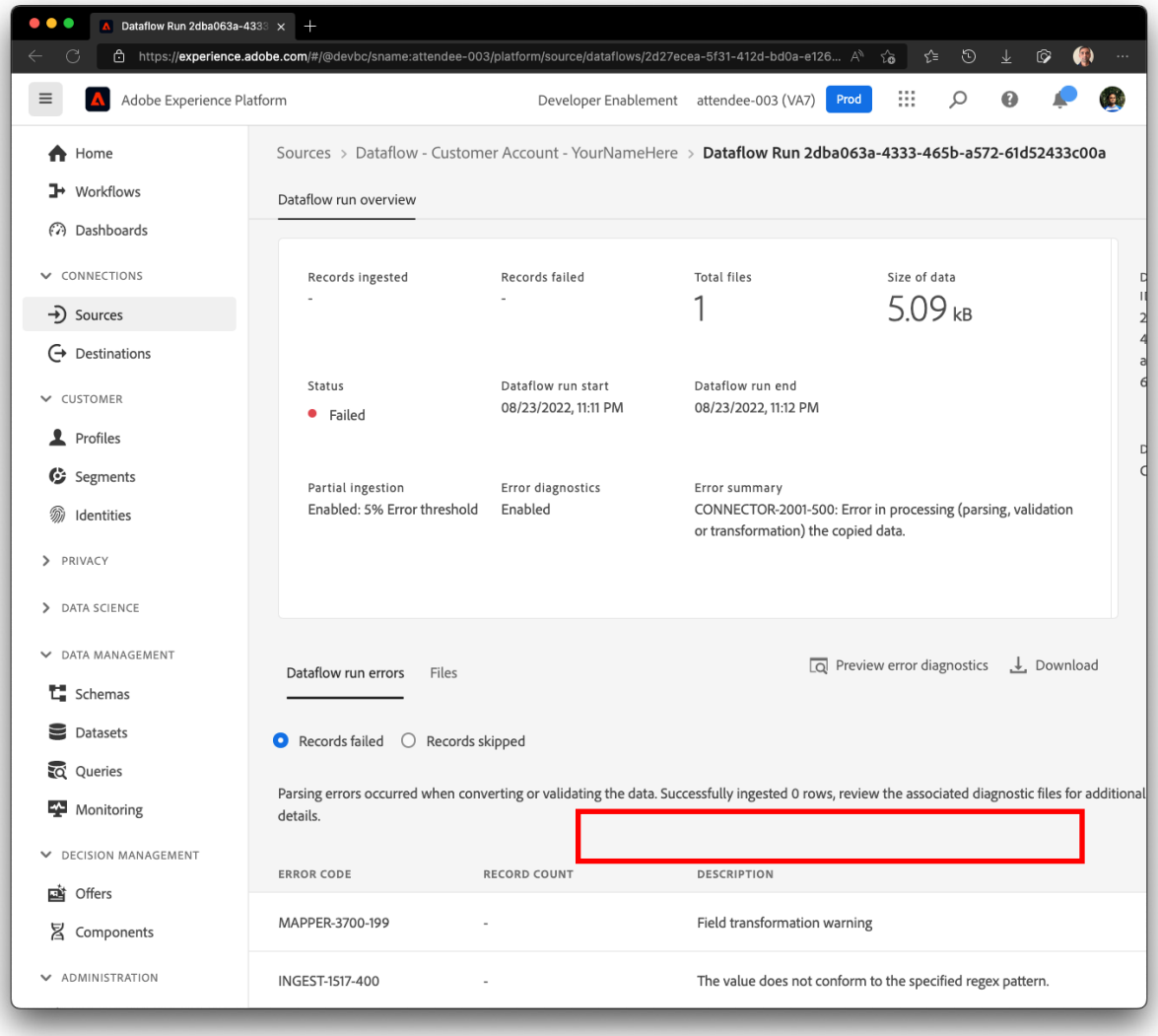


We will notice that **INGEST-1517-400** error code is listed.

ERROR CODE	RECORD COUNT	DESCRIPTION
MAPPER-3700-199	-	Field transformation warning
INGEST-1517-400	-	The value does not conform to the specified regex pattern.



Click on the Preview error diagnostics to preview the error messages in detail.



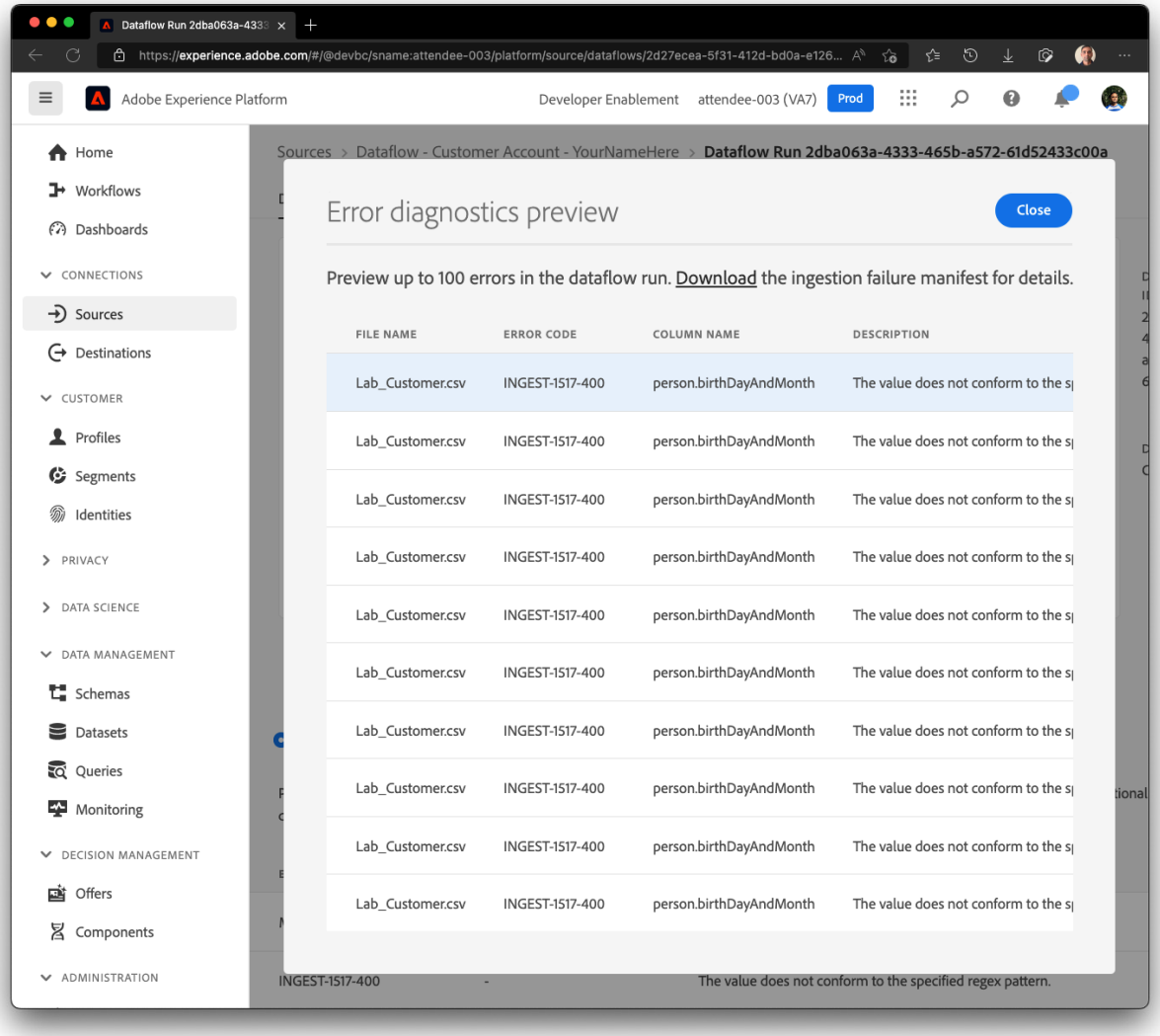
Error diagnostics preview will load. We can see the XDM attribute where the error occurred (**COLUMN NAME**) and the actual error message (**DESCRIPTION**).

Error 1

The description will have the following text

The value does not conform to the specified regex pattern: [0-1][0-9]-[0-9][0-9] in field: person.birthDayAndMonth of type: String.

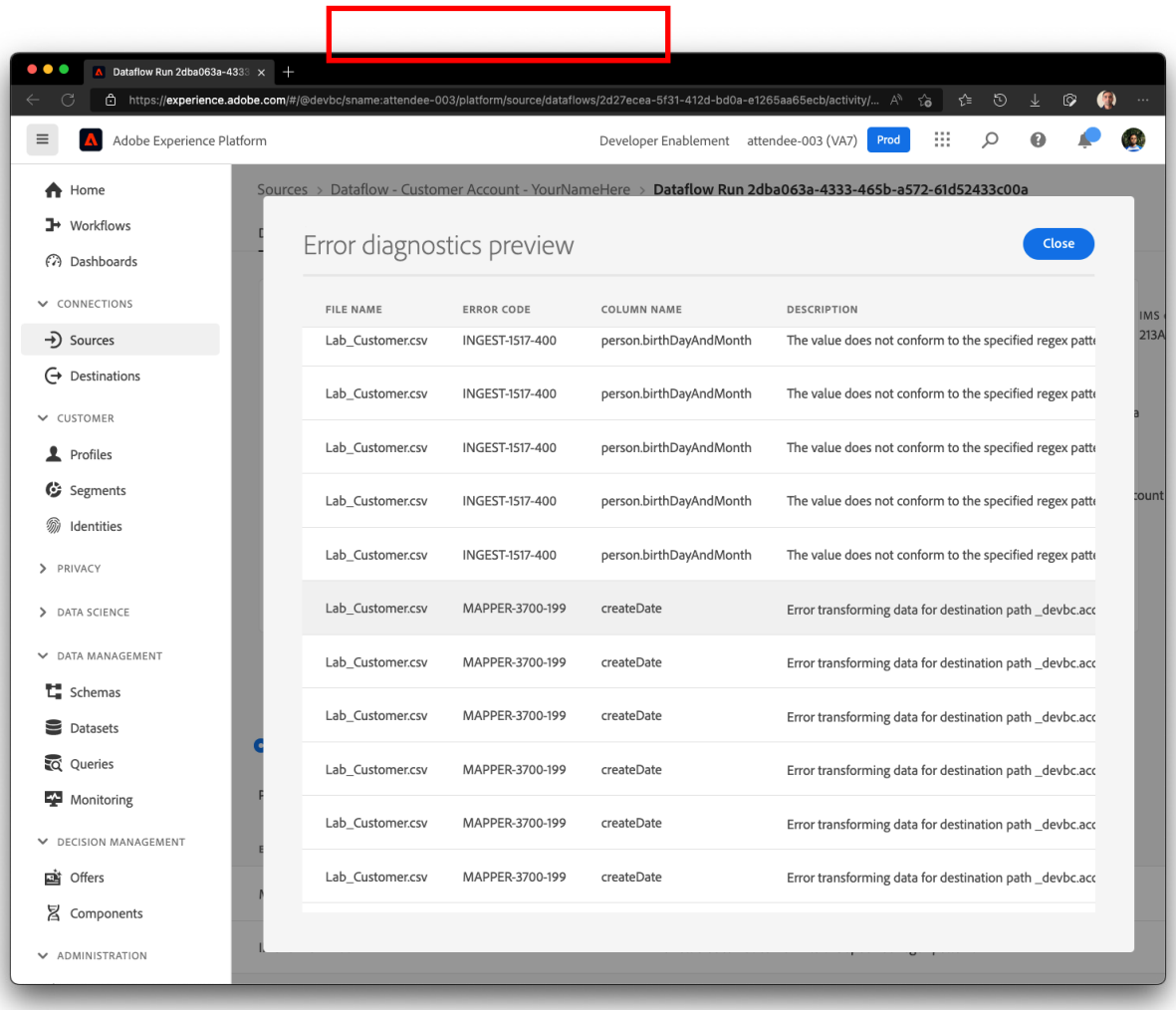
This error occurs because **person.birthDayAndMonth** is expected in the format of two digit month followed by a hyphen followed by two digit day. For example, the values generated are 4-27. Whereas, the values should be 04-27.



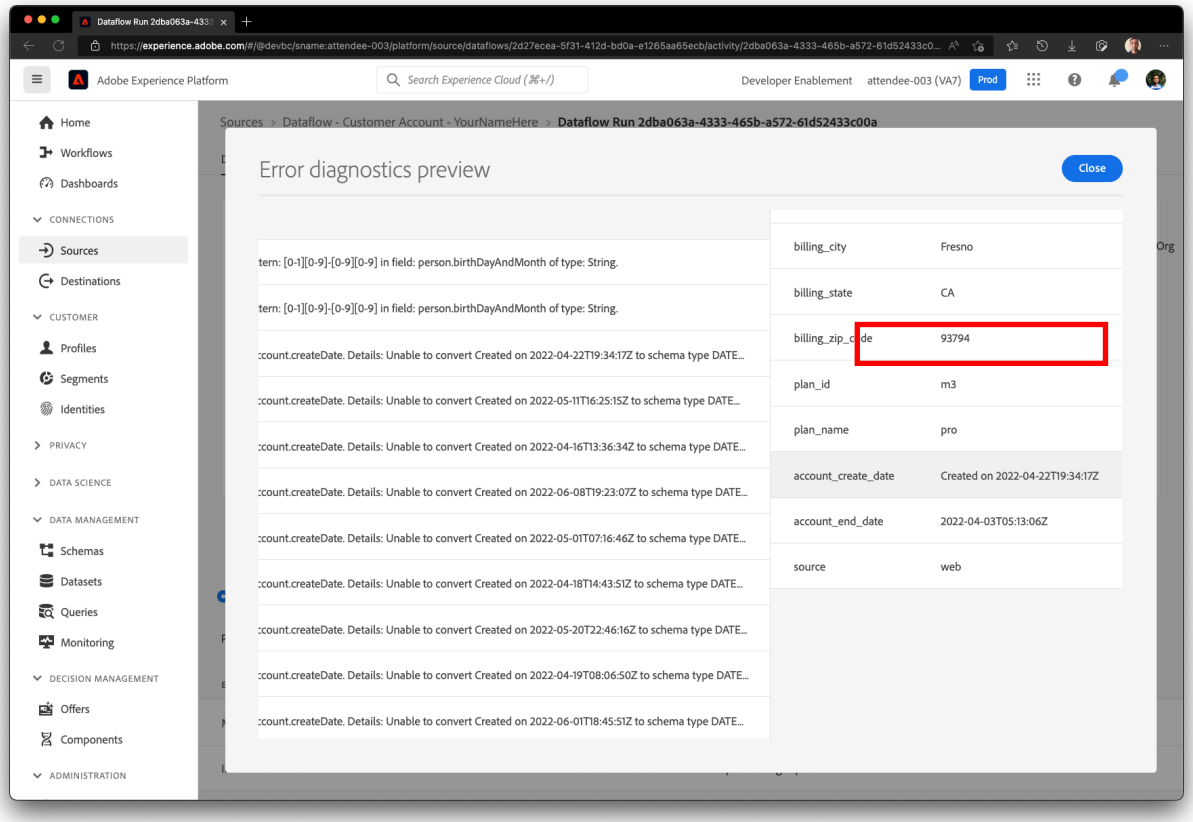
Error 2

Scroll down to the see the second error message associated with **createDate**. The description will have the following text

Error transforming data for destination path _devbc.account.createDate. Details: Unable to convert Created on 2022-04-22T19:34:17Z to schema type DATE...



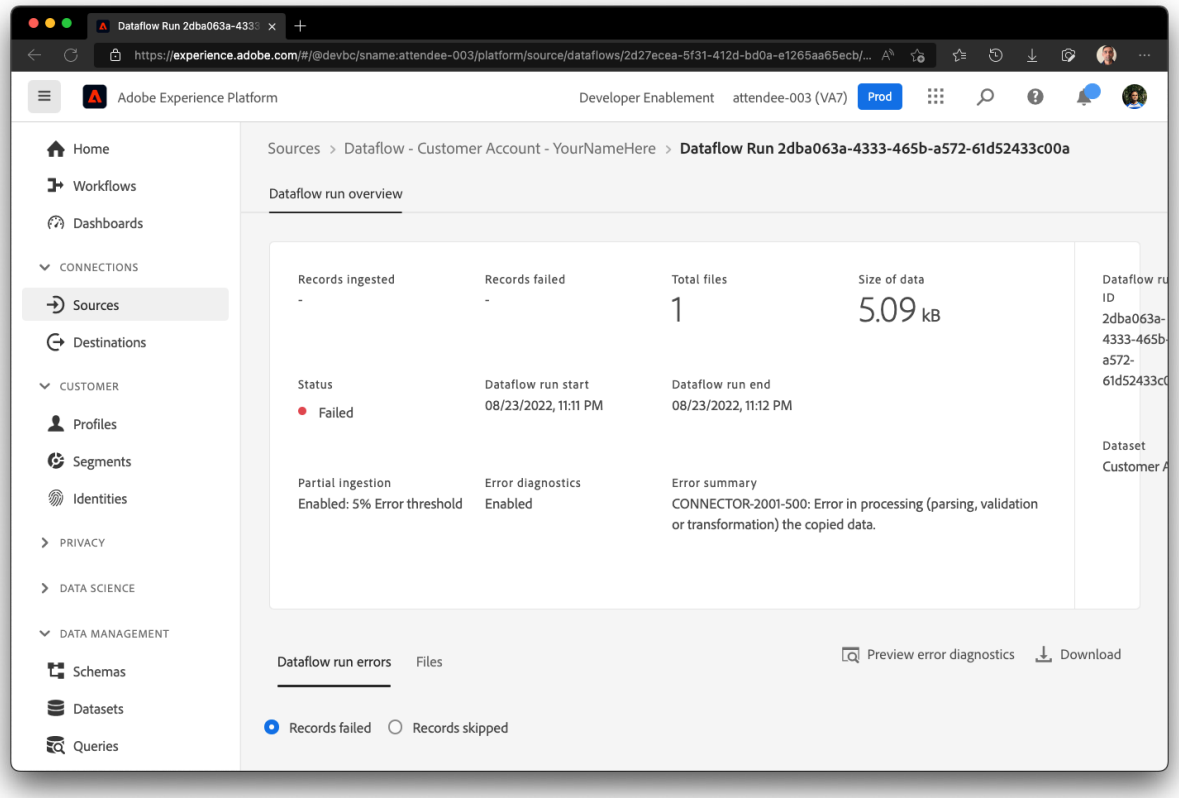
This error occurs because the source data has strings such as "**Created on 2022-04-22T19:34:17Z**". This value cannot be converted into a Date automatically. A calculated field must be used to cleanse the data. Scroll to the right side to see the source data associated with this error code.



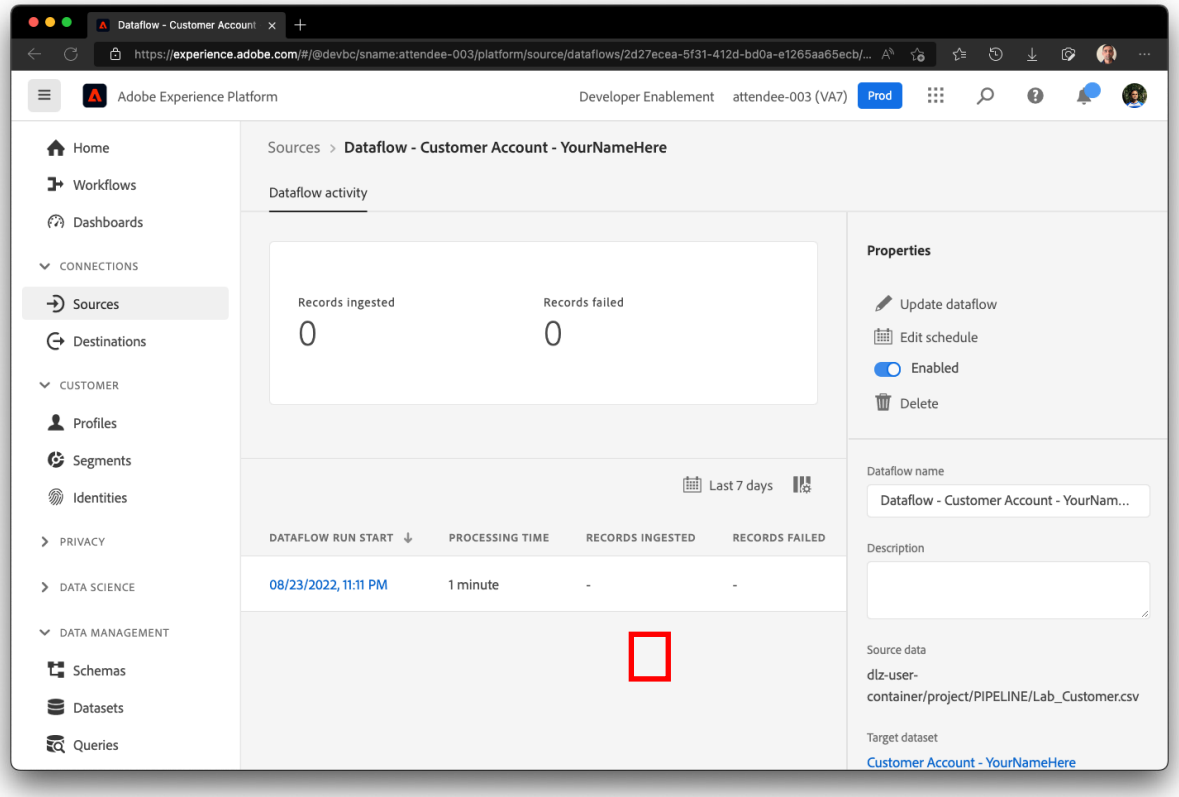
We will now go back and fix these errors.

3.9. Correct the Dataflows

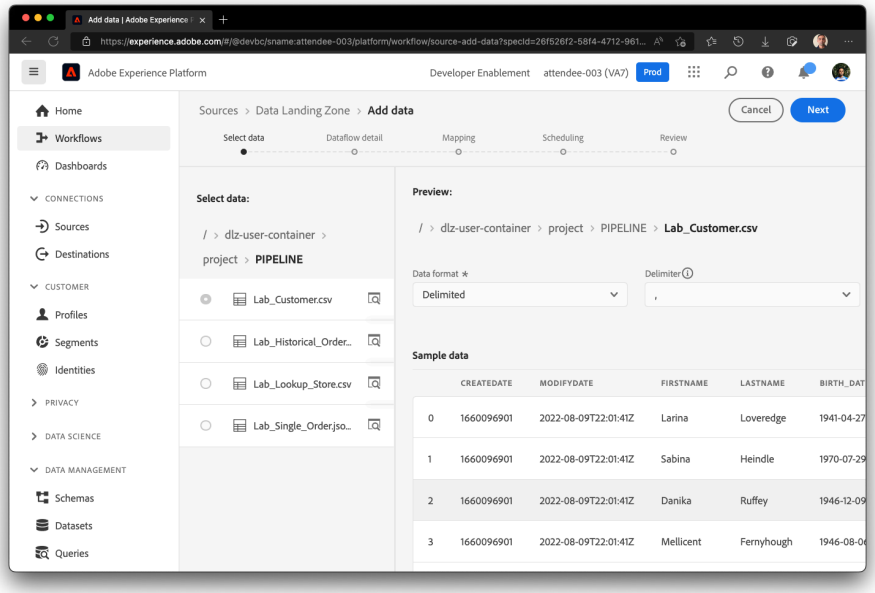
Go back to the Dataflows page by clicking on the Dataflow name.



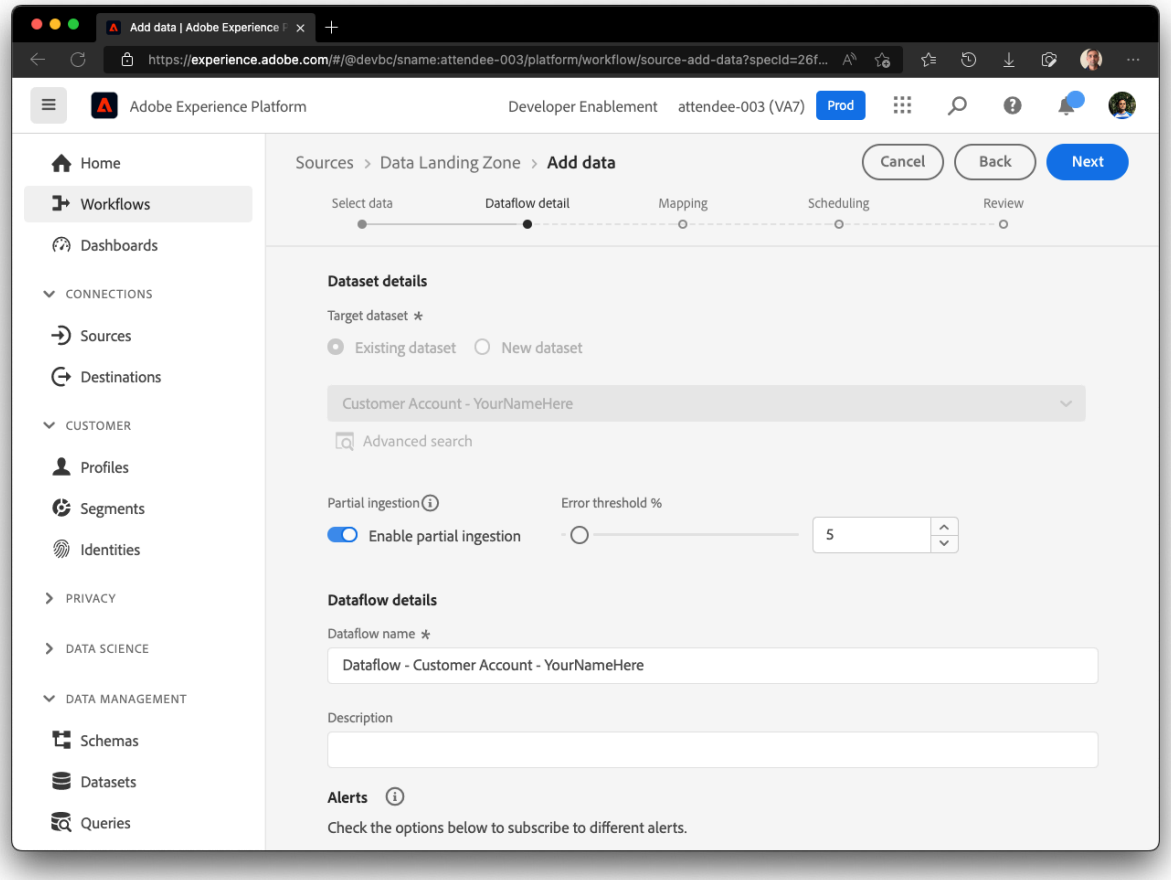
Click on the Update dataflow



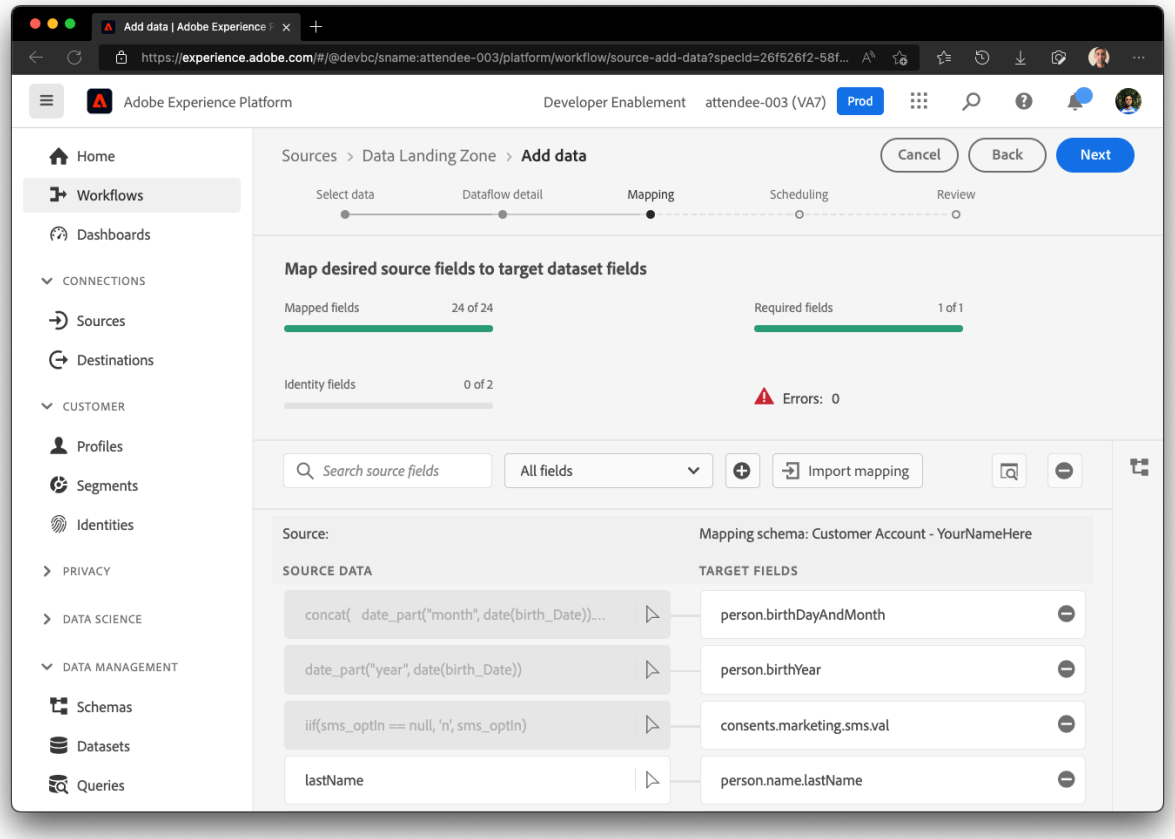
Data Preview screen will load. Click Next to go to Dataflow detail page.



Click Next to proceed to Data Prep Mapper step

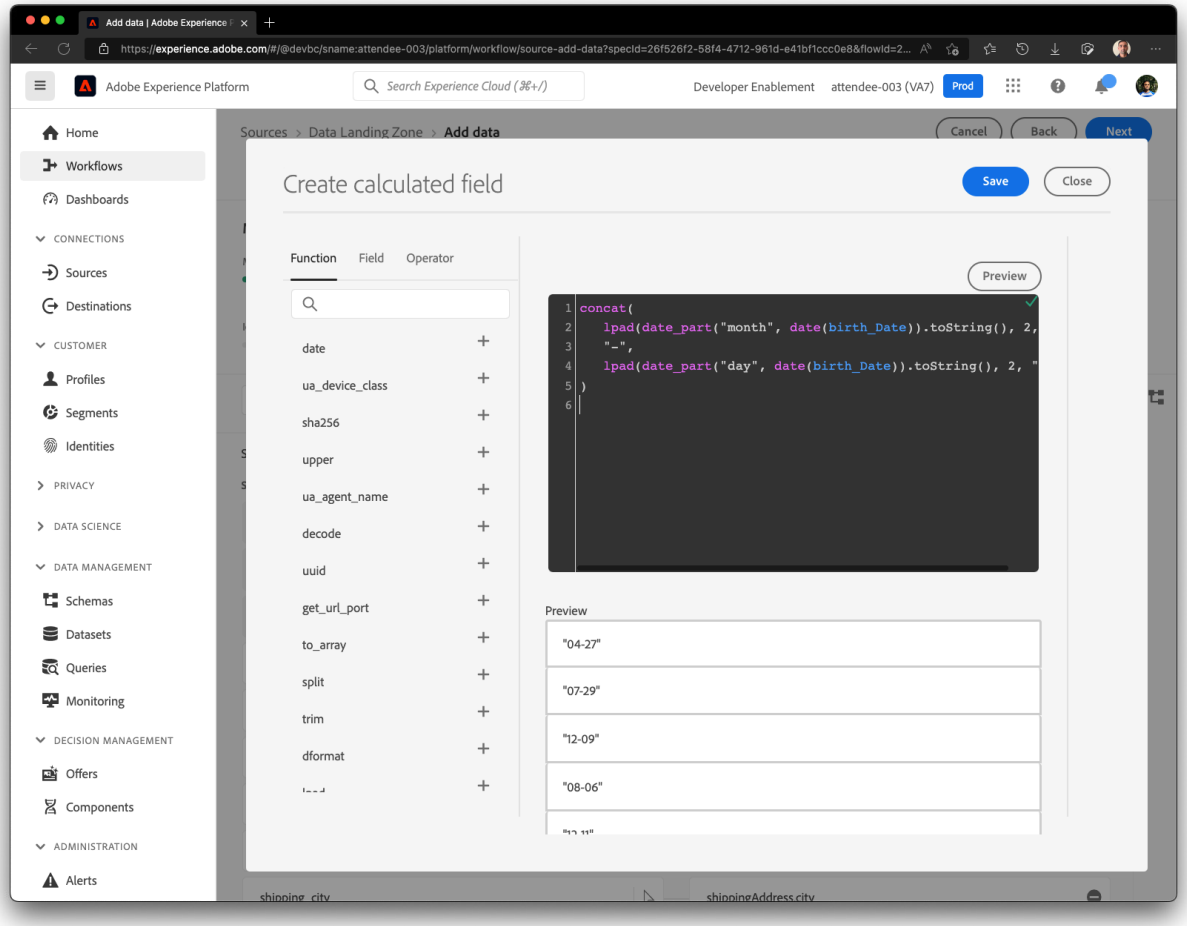


Data Prep Mapper step will load. Click on the arrow icon next to the calculated field populating the **person.birthDayAndMonth** XDM attribute.

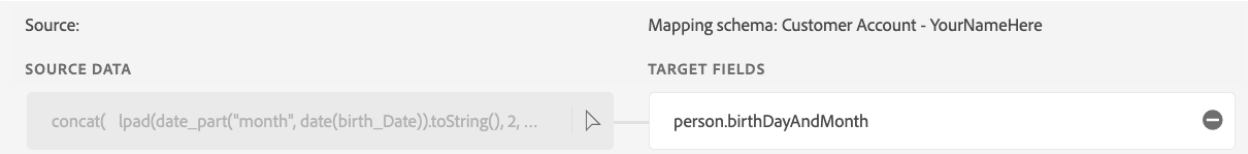


Type in the following expression in the calculated field and click Preview. Data will appear as 2 digit month and 2 digit day

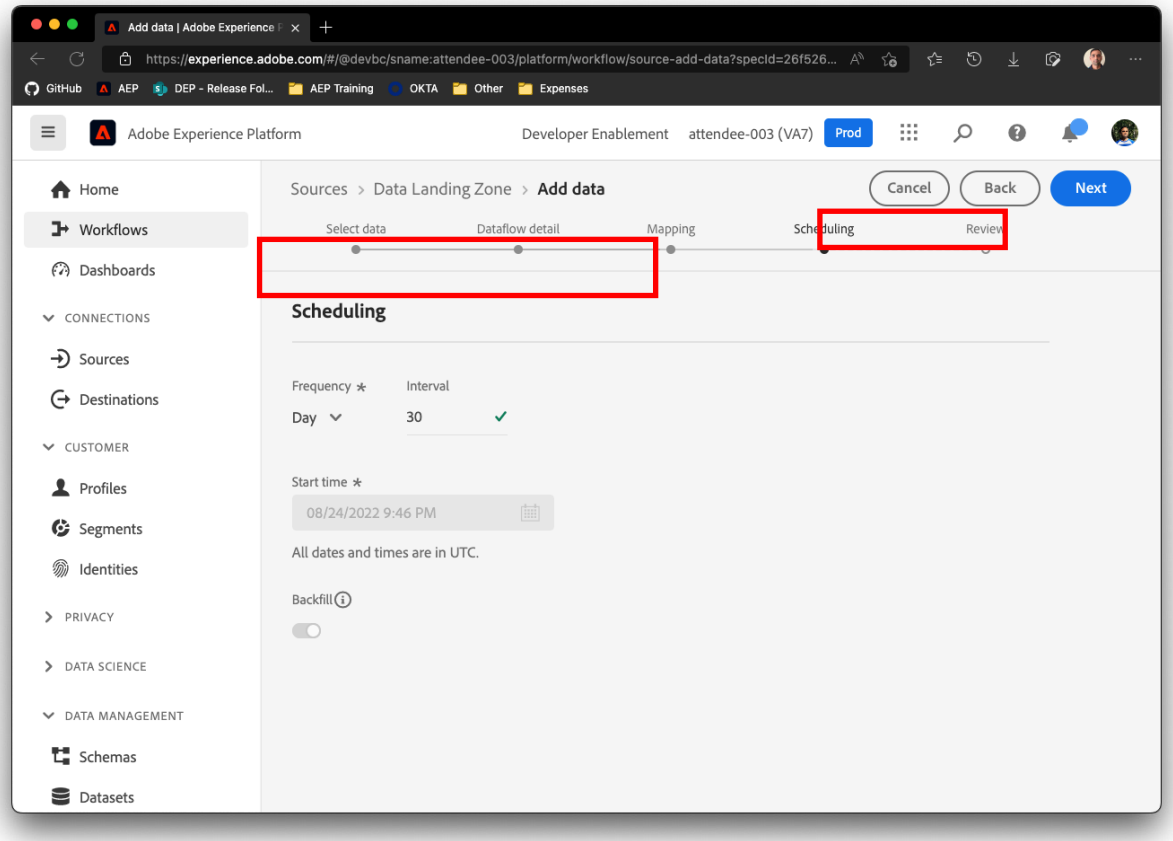
```
concat(
  lpad(date_part("month", date(birth_Date)).toString(), 2, "0"),
  "-",
  lpad(date_part("day", date(birth_Date)).toString(), 2, "0")
)
```



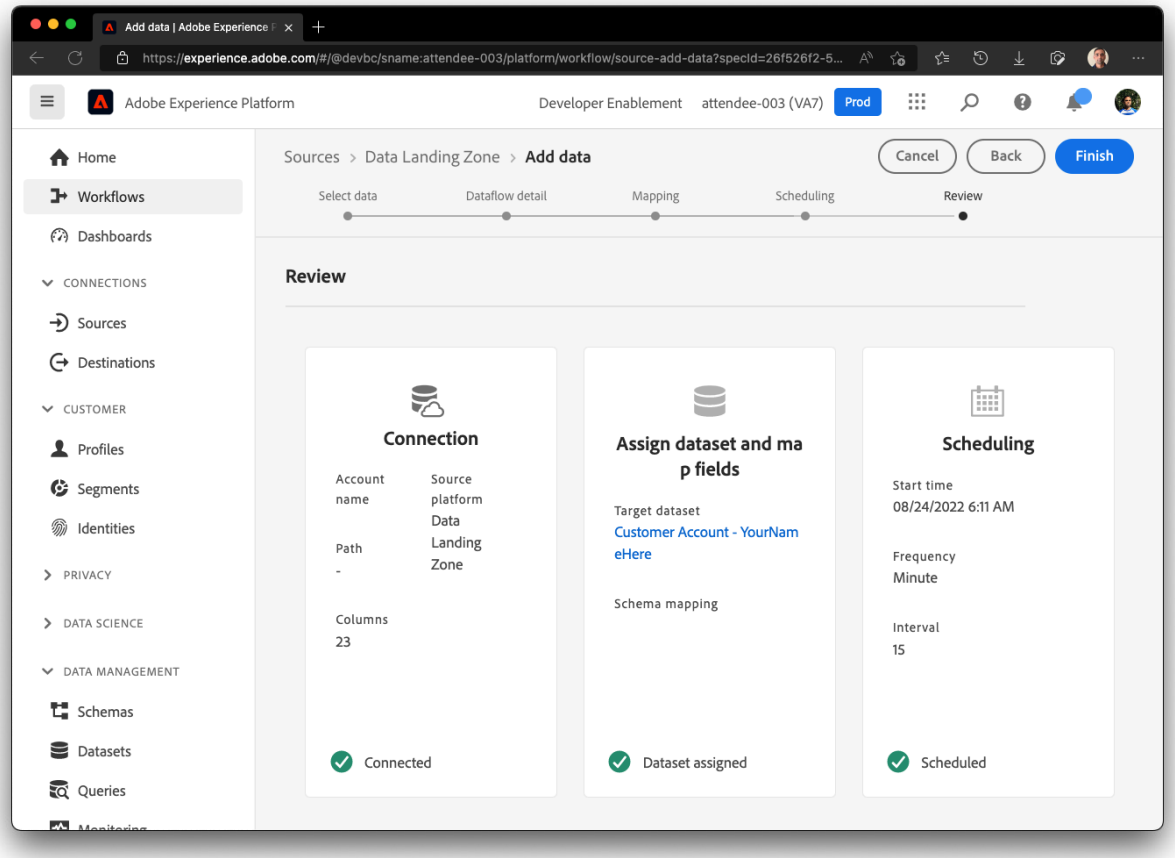
Click save. The mapping will look like this



Leave the schedule as 30 days and click next to continue.



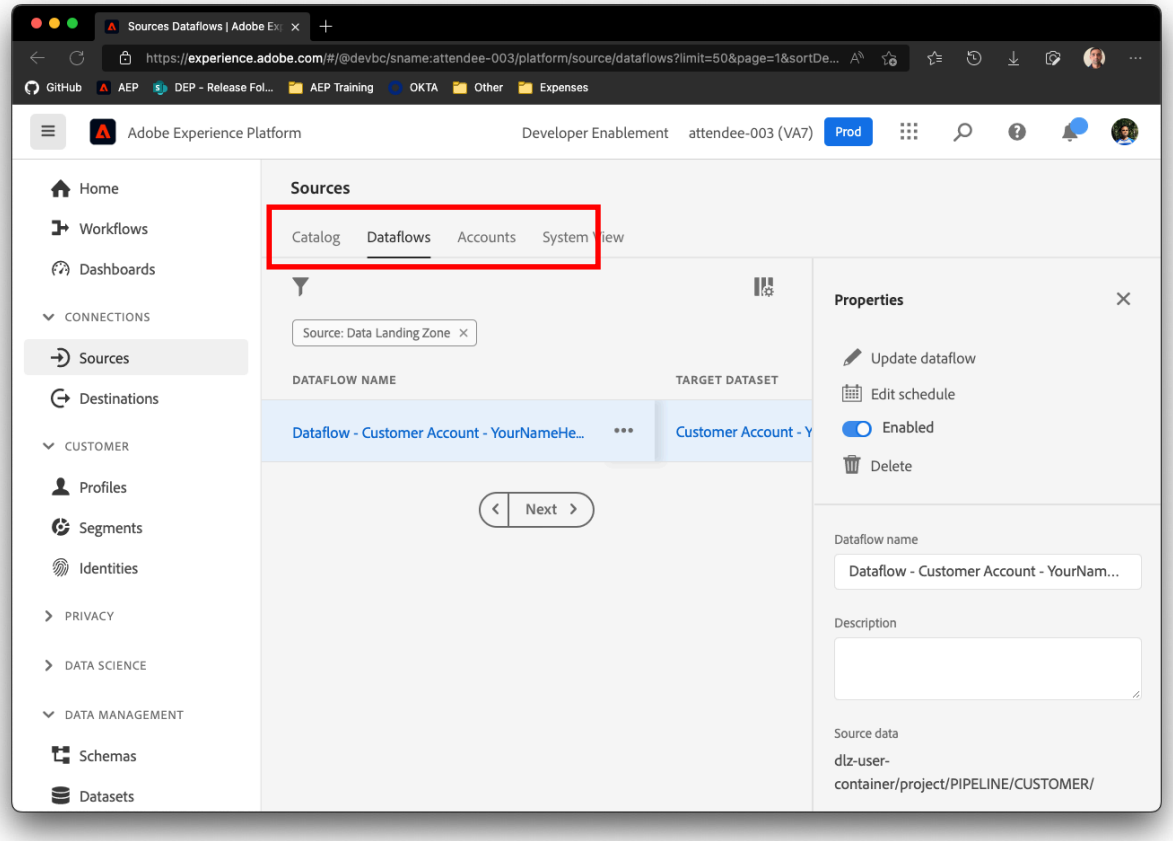
Click Next to review the dataflow and then finally save the dataflow by clicking Finish



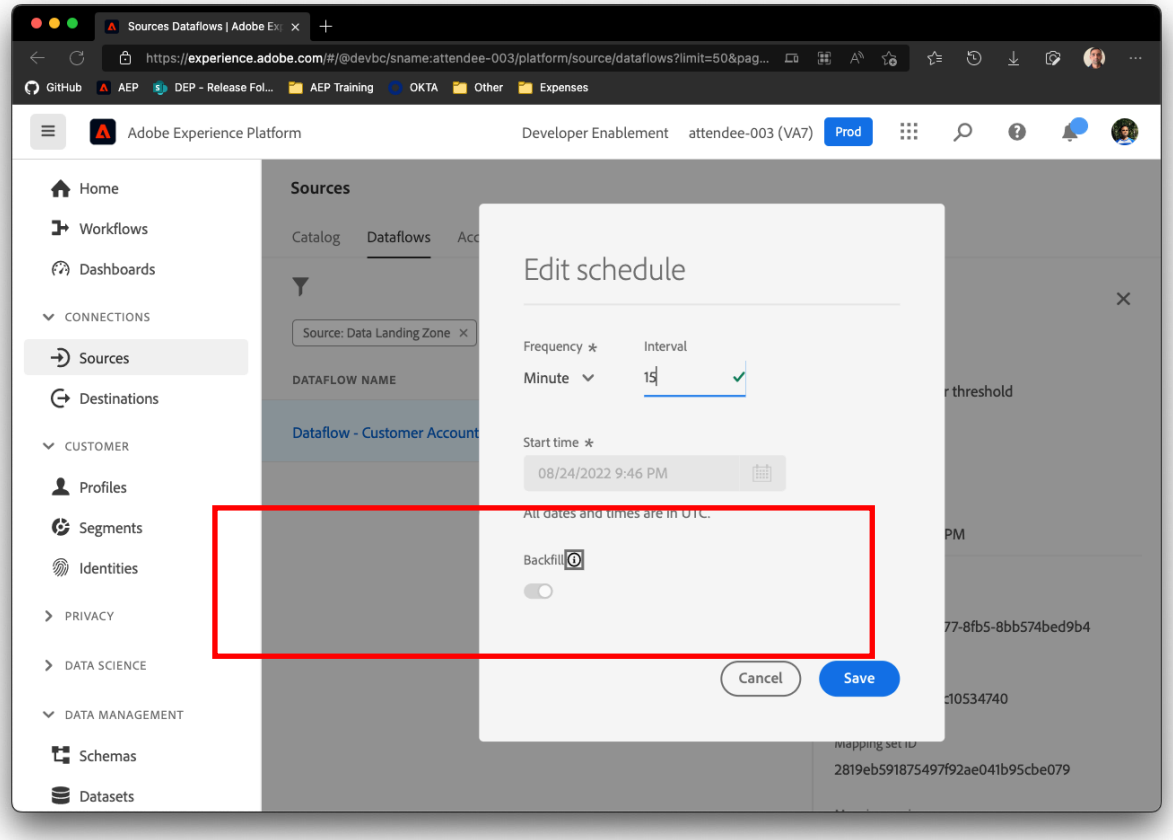
NOTE	The changes we made to Data Prep step will be applicable only for the future runs. It will NOT affect the previous runs.

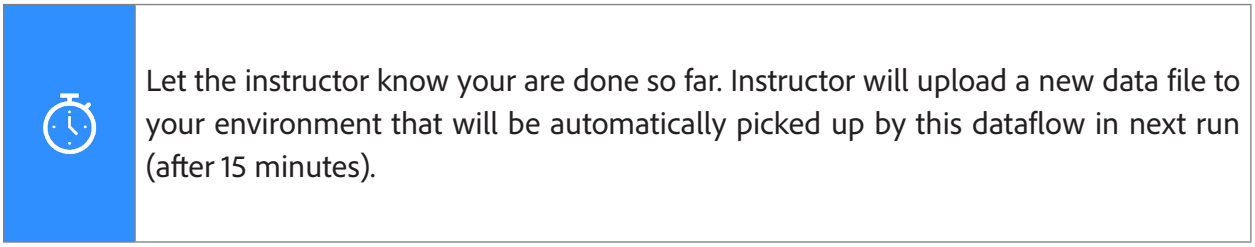
3.10. Re-schedule the failed Dataflow

In the Dataflow page, click on the row to select the dataflow, when the right panel opens up, click **Edit schedule**.



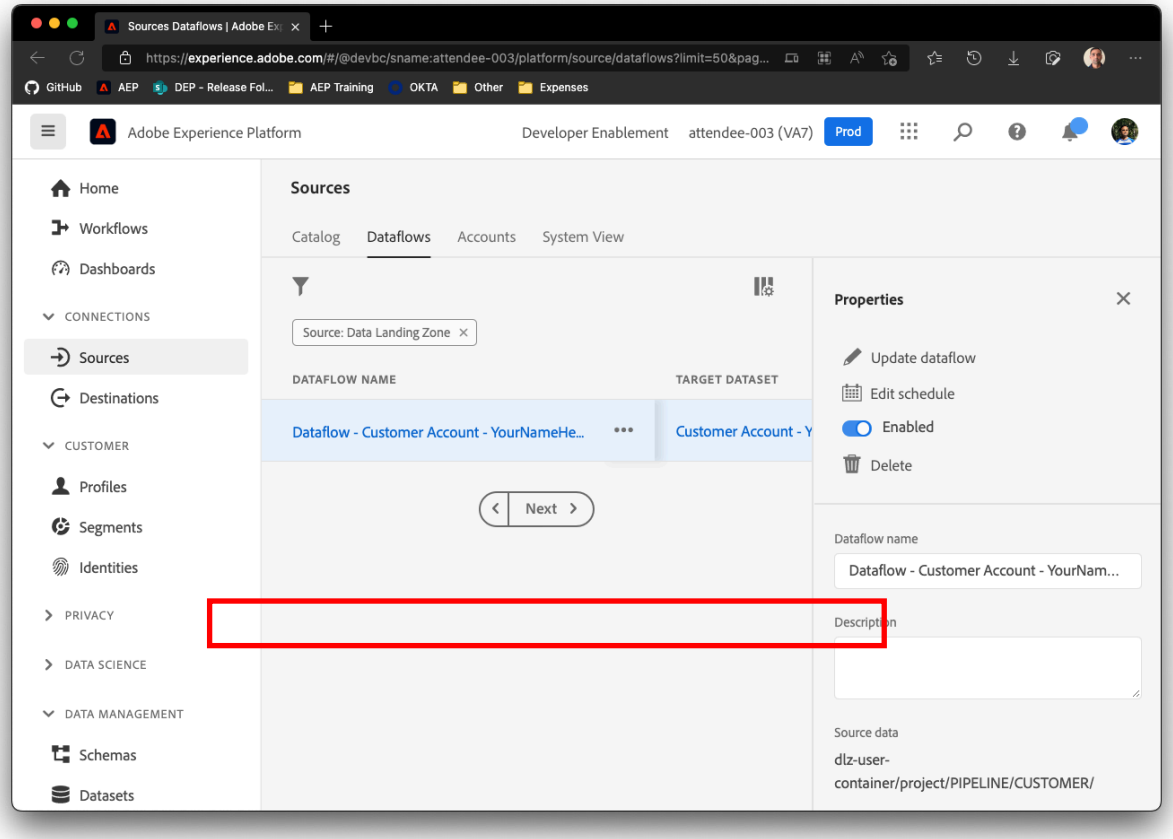
Change the schedule such that Frequency is set to **Minute** and Interval is set to **15** as shown below



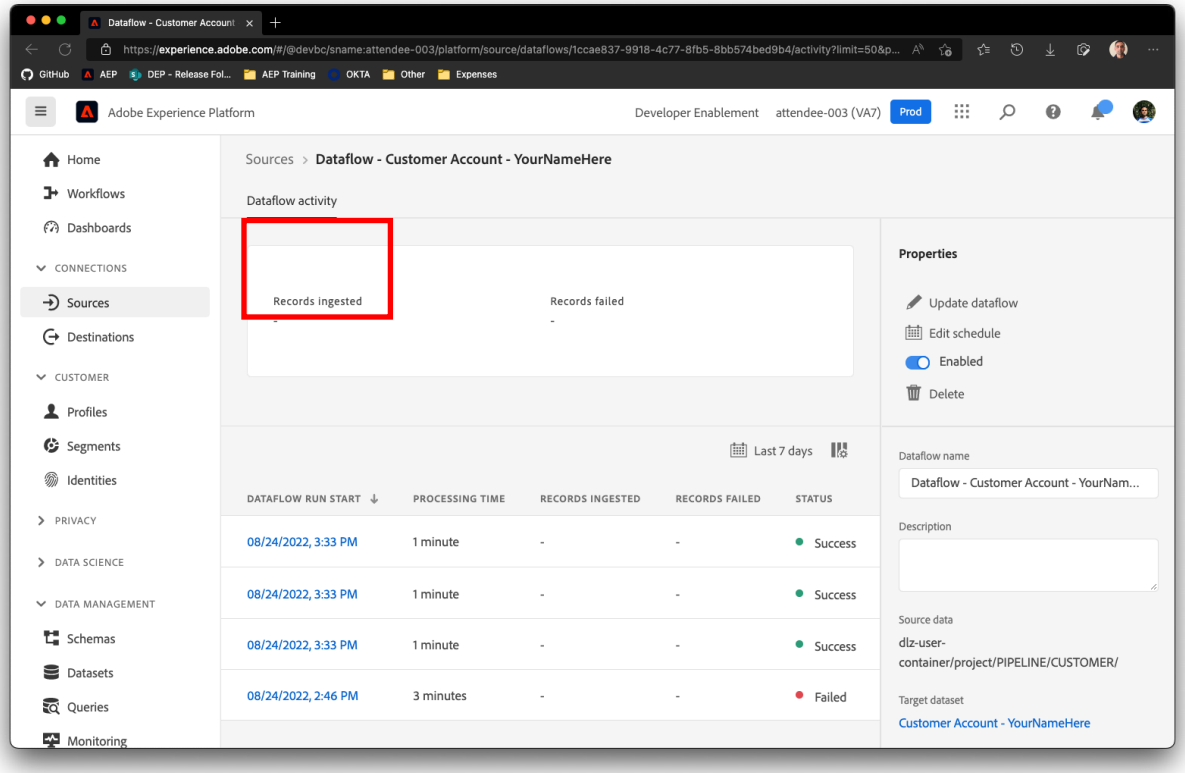


Let the instructor know your are done so far. Instructor will upload a new data file to your environment that will be automatically picked up by this dataflow in next run (after 15 minutes).

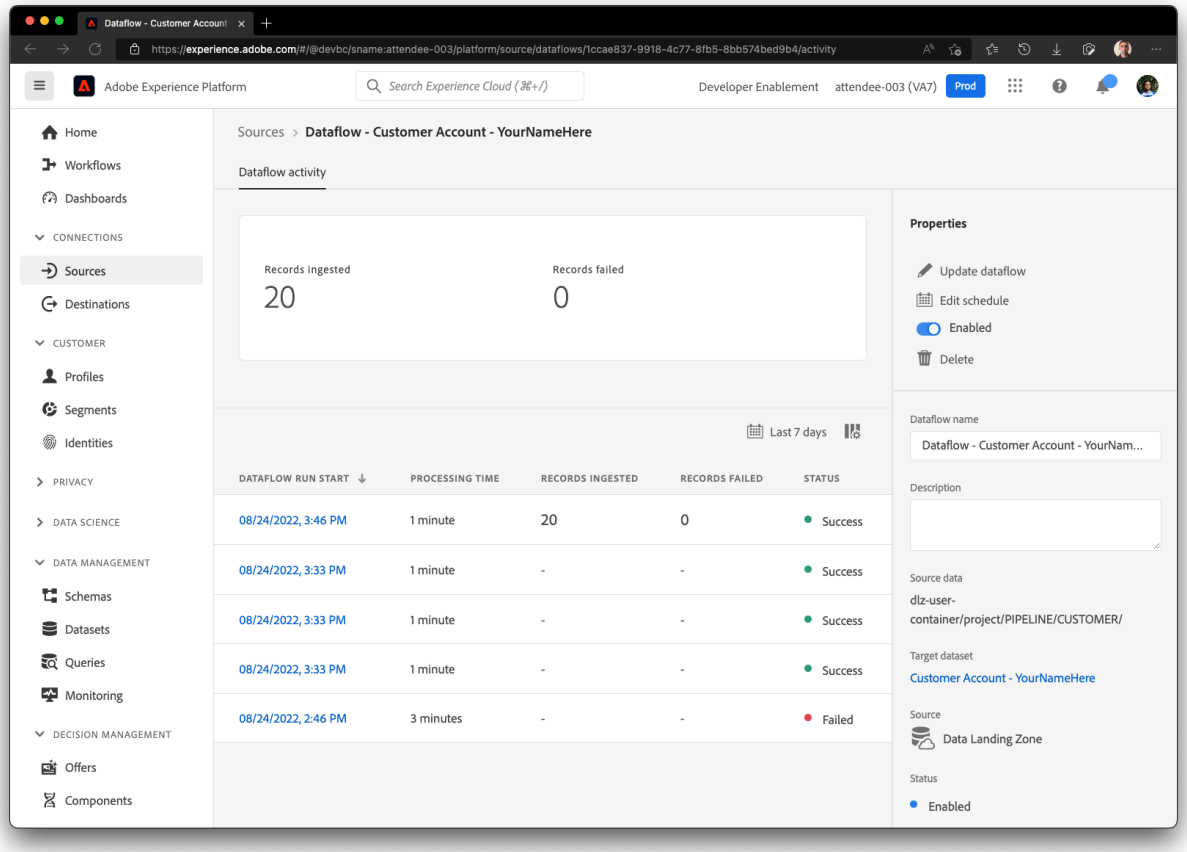
Click on the Dataflow name on the screen to open Dataflow runs



You will see some Dataflow runs with Success status and NO records ingested as shown below.



After the instructor uploads a new datafile, you will notice a successful run that ingested the data. You should notice 20 records ingested. Click on the date timestamp for this Dataflow run



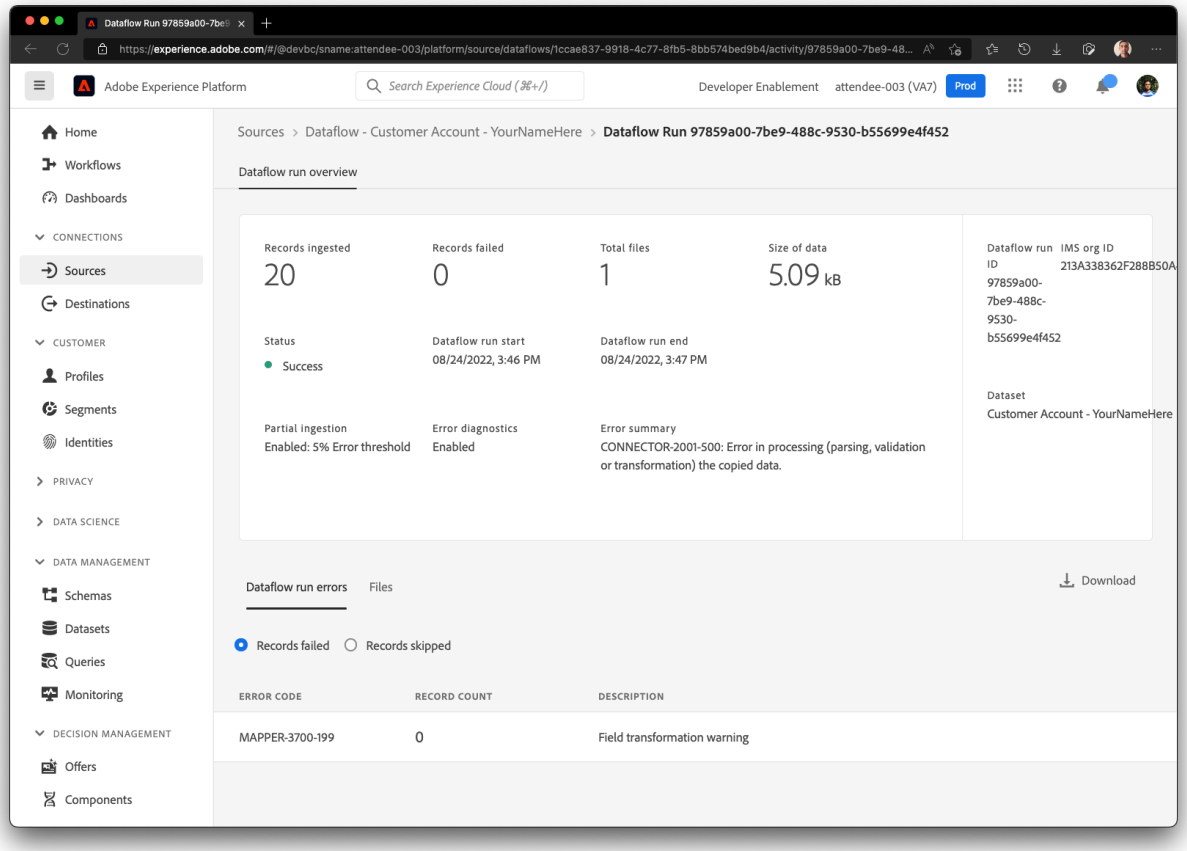
We will notice that 20 records are now ingested into the Dataset.

Adobe Experience Platform
Lab 3 - Data Pipeline Historical

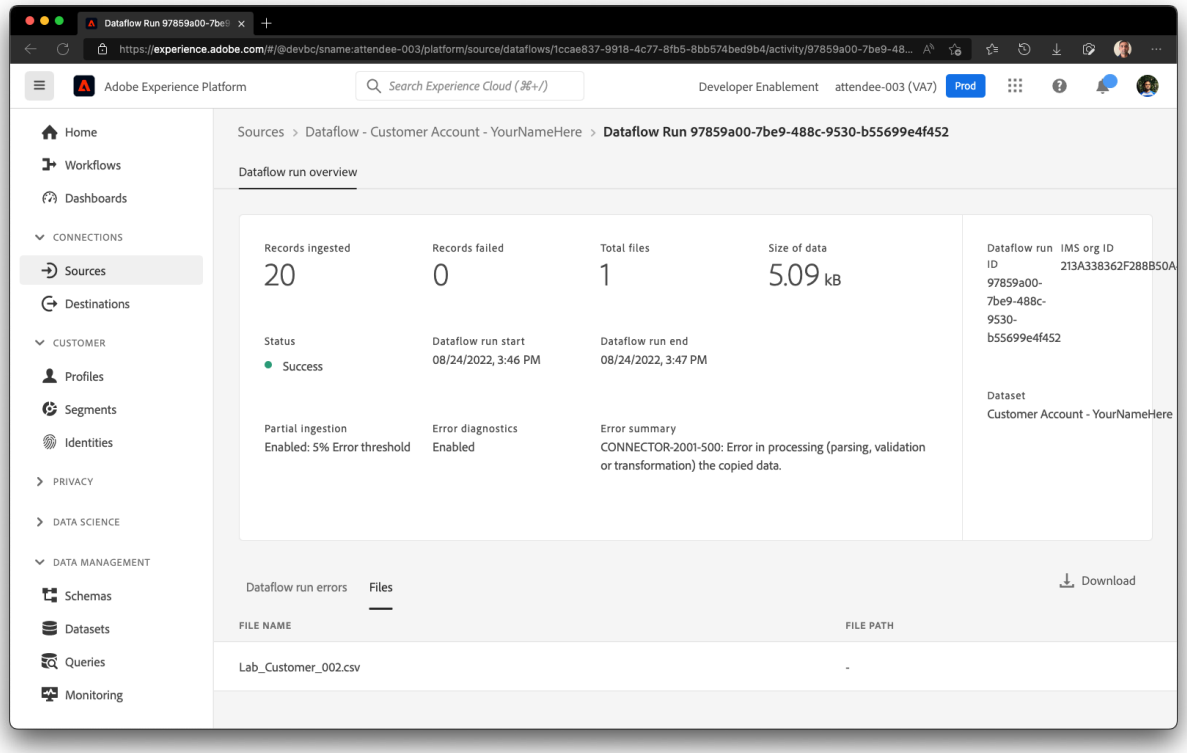
Adobe Experience Platform Bootcamp Deep Dive Edition

Name _____

Sandbox _____



Click on the Files tab and confirm the Lab_Customer_002.csv got loaded.

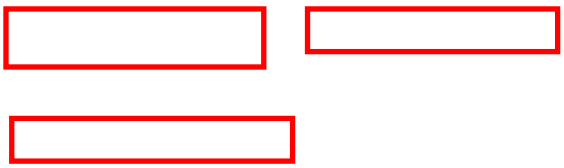


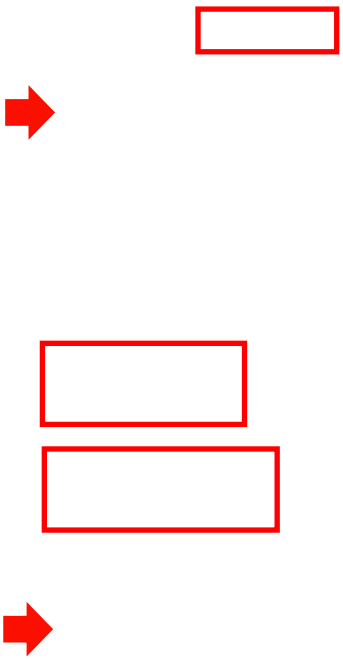
The lab is now complete.

Learnings

In the second part of the lab, we learned:

1. Editing an existing Dataflow to make corrections to transformation logic
2. Change the schedule of an existing Dataflow
3. Check the files loaded in a given Dataflow run using Monitoring console.





1. Lab Overview

Ingest Orders historical data into AEP in batch mode using Data Landing Zone (this lab) and stream orders live data using Streaming Source HTTP API (next lab).

This lab will introduce some complex data transformations and JSON data processing

Expected time: 60 minutes

2. Learning Objectives

What should you walk away with after taking this Lab?

- Understand ingestion of JSON data
- Use Data Landing Zone as a source
- Use Data Prep to Map the non-XDM data to XDM
- Scheduling batch workflows

3. Lab Tasks – Orders – Historical data

In this exercise, we will load the Orders data from Data Landing Zone to AEP Data Lake and Profile.

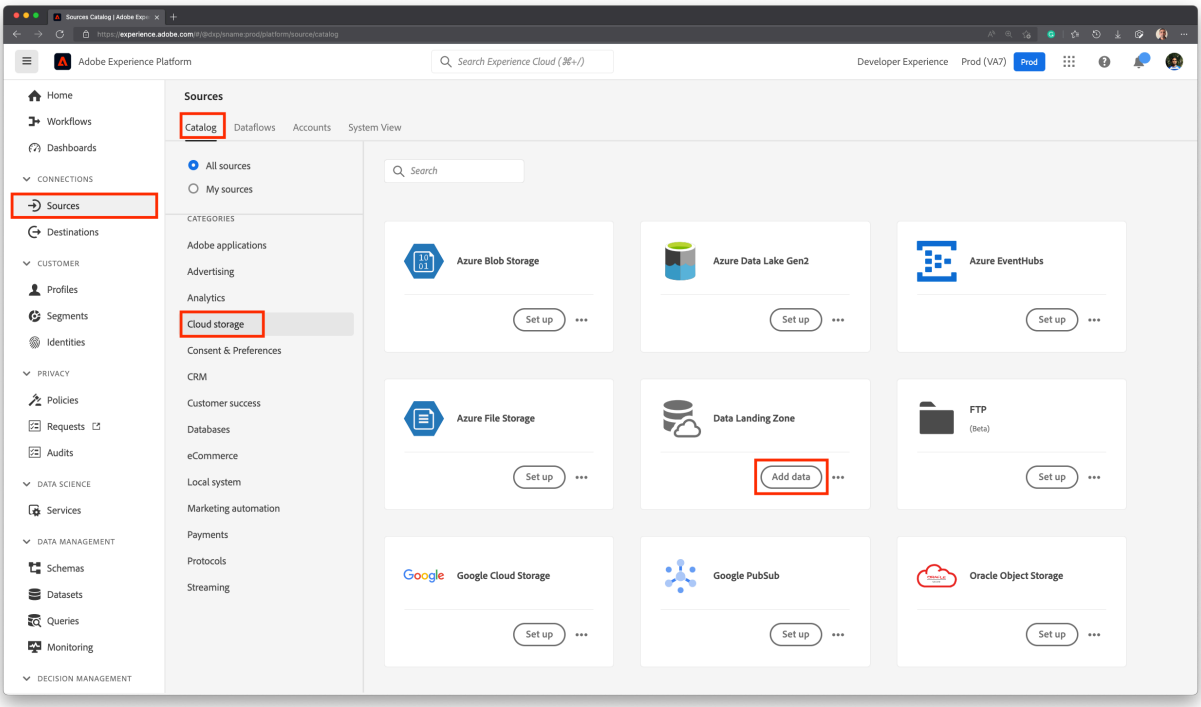
Pre-requisites

1. Orders JSON file uploaded in the Azure ADLS Directory
2. Orders Schema and Dataset are already created

Steps

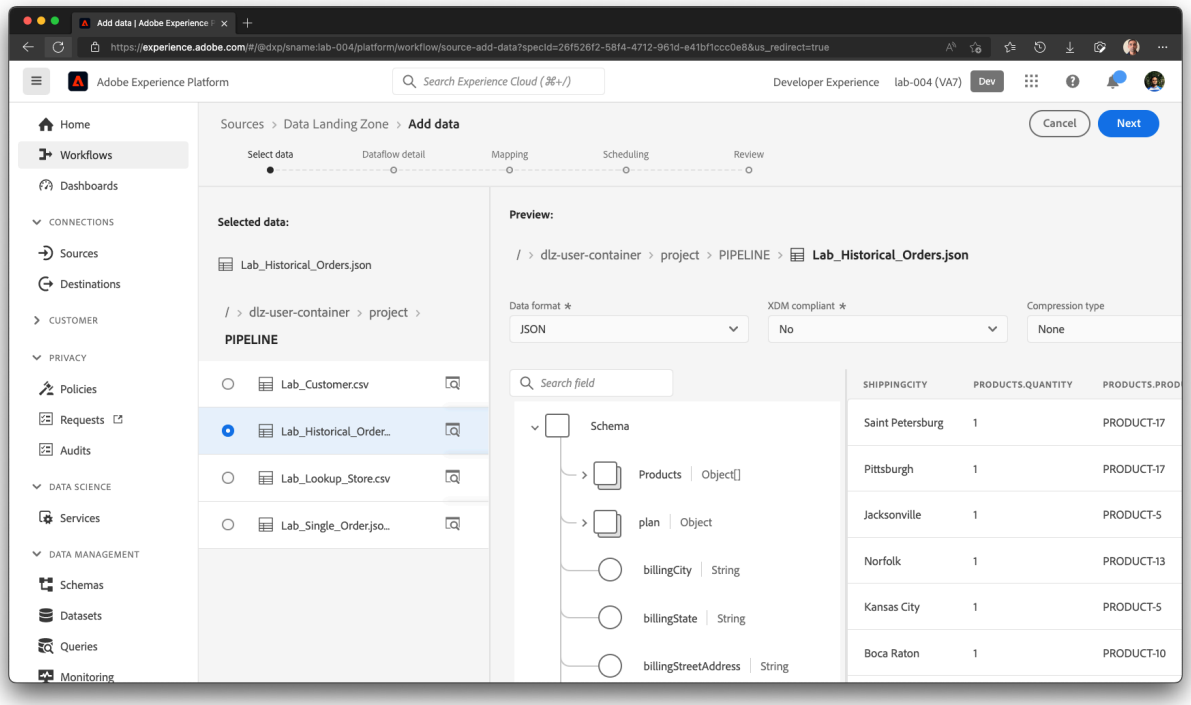
Go to Adobe Experience Platform à **Sources** à **Catalog** à **Cloud storage**. Click on **Setup** / **Add Data** for the Data Landing Zone.

Tip	If at least one connection exists for that source, you will see "**Add data**" as the default action. If no connections exist for that source, you will see "**Setup**" as the default action.



3.1. Select Source data

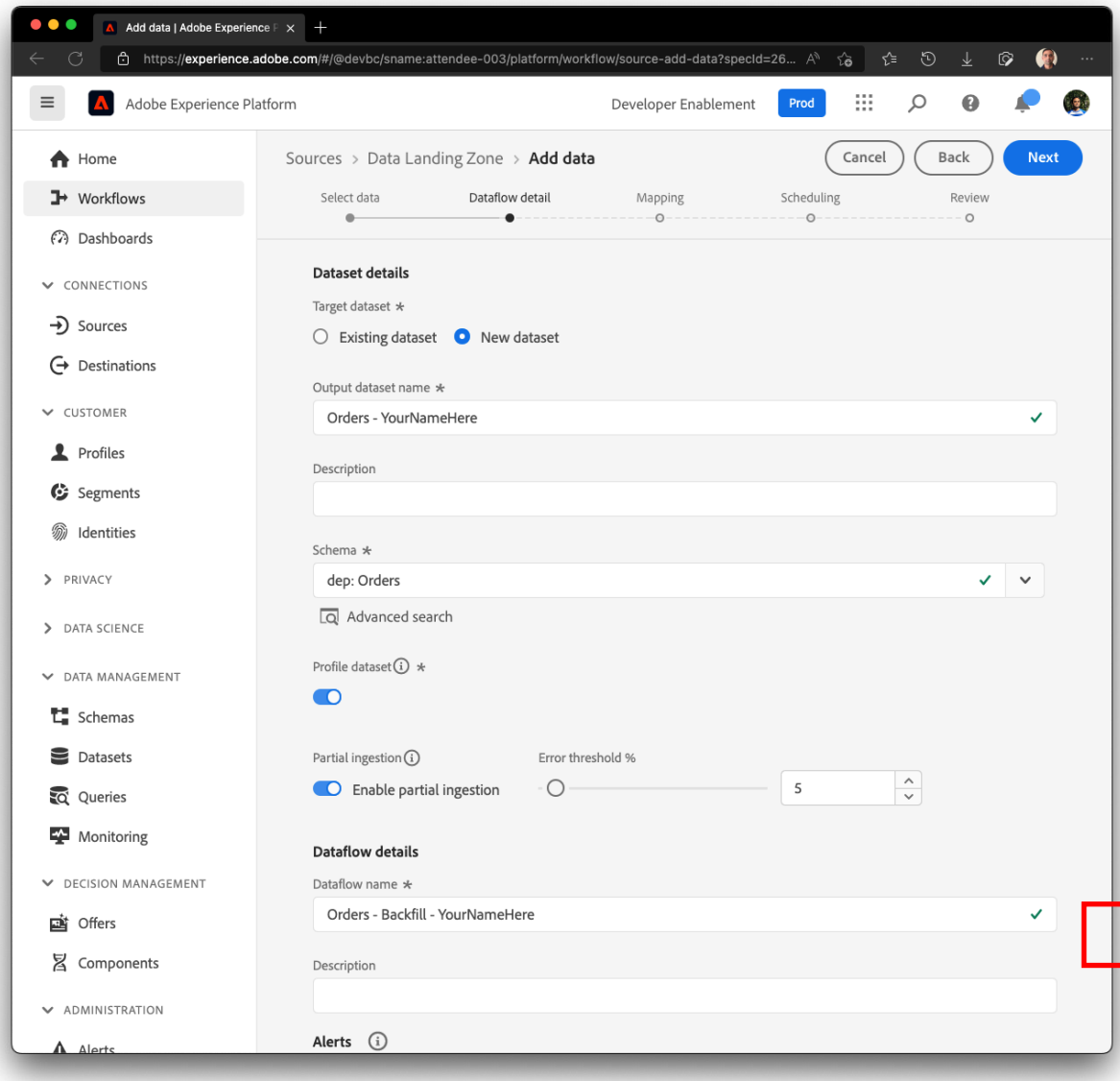
In the **Select data** screen, navigate to **dlz-user-container** → **project** → **PIPELINE** and choose **Lab_Historical_Orders.json**. On the right-hand side, choose data format as **JSON.** Preview of the selected file is automatically displayed. Click **Next**.



3.2. Define the Target dataset

In the **Dataset details** step, choose ***New dataset*** option. Name the output dataset as **Orders - YourNameHere**. Set the schemas as **dep: Orders**.

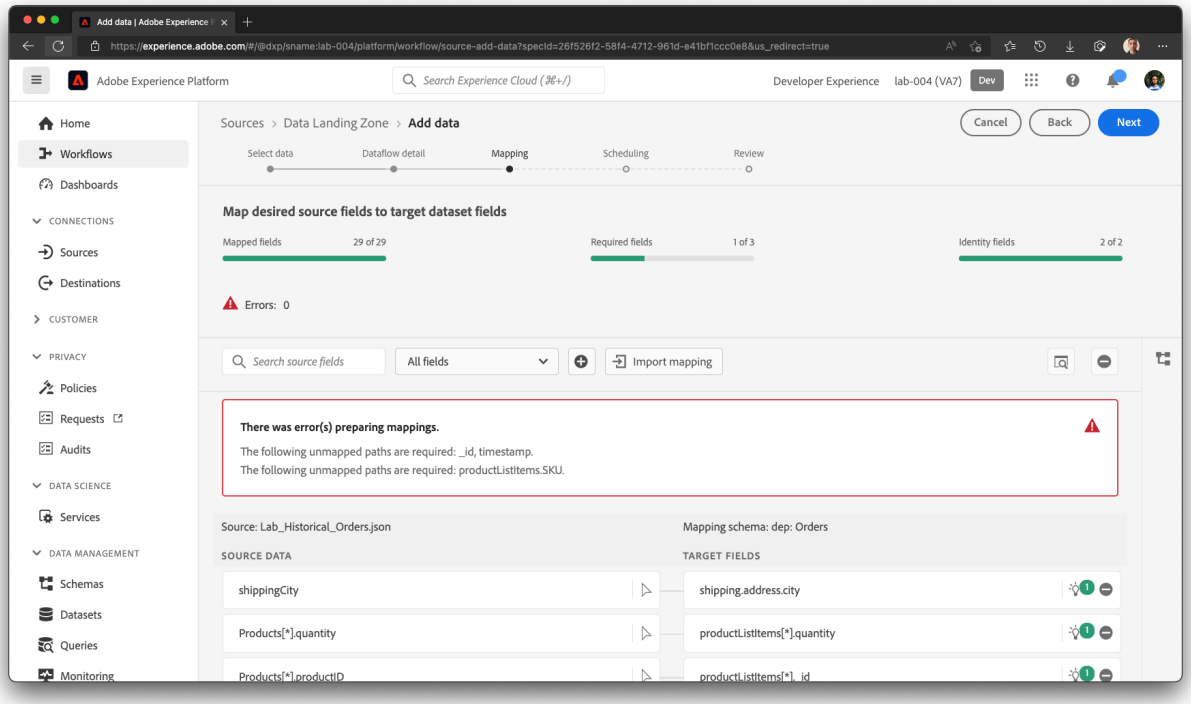
Turn ON the **Profile dataset** option. Enable **Partial ingestion**. Name the Dataflow as **Dataflow – Orders – Backfill**.



Tip	When Partial Ingestion is enabled, Error diagnostics are automatically enabled and hence the toggle box will disappear.

3.3. Data Prep / Transformation

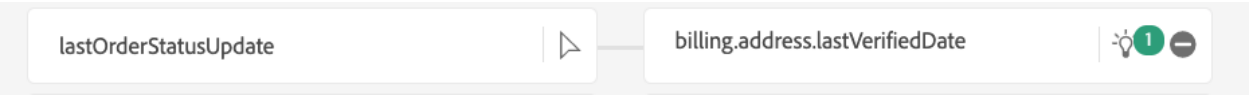
In the Data Prep (Mapping) step, ML Recommendations will automatically map most attributes. However, you will also see several errors. The initial screen will look similar to below. **_id** and **timestamp** are never recommended or mapped by default for Experience Events.



Please ensure the mappings are accurate. The initial recommendations loaded may have the following **invalid** mappings

3.4. Remove the following mappings

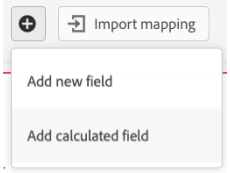
- Mapping to XDM attribute **billing.address.lastVerifiedDate**



3.5. Create Calculated Fields

Mapping to _id

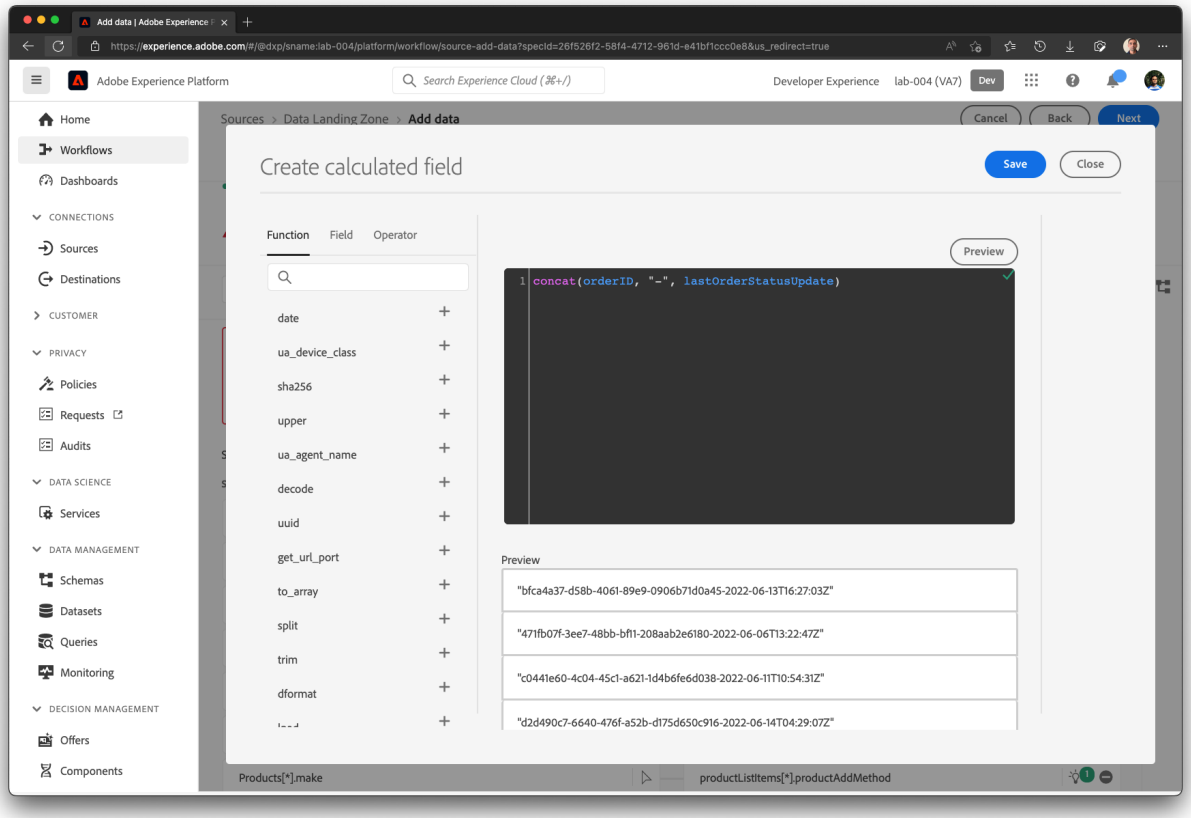
Create the following calculated fields by clicking on  and click on Add Calculated Field.



Write the following expression and click Preview

```
concat(orderID, "-", lastOrderStatusUpdate)
```

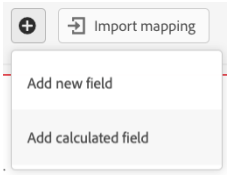
Calculated field will look similar to this. Click Save to save the calculated field.



Map the calculated field to **_id**

Mapping to order._devbc.acqSource

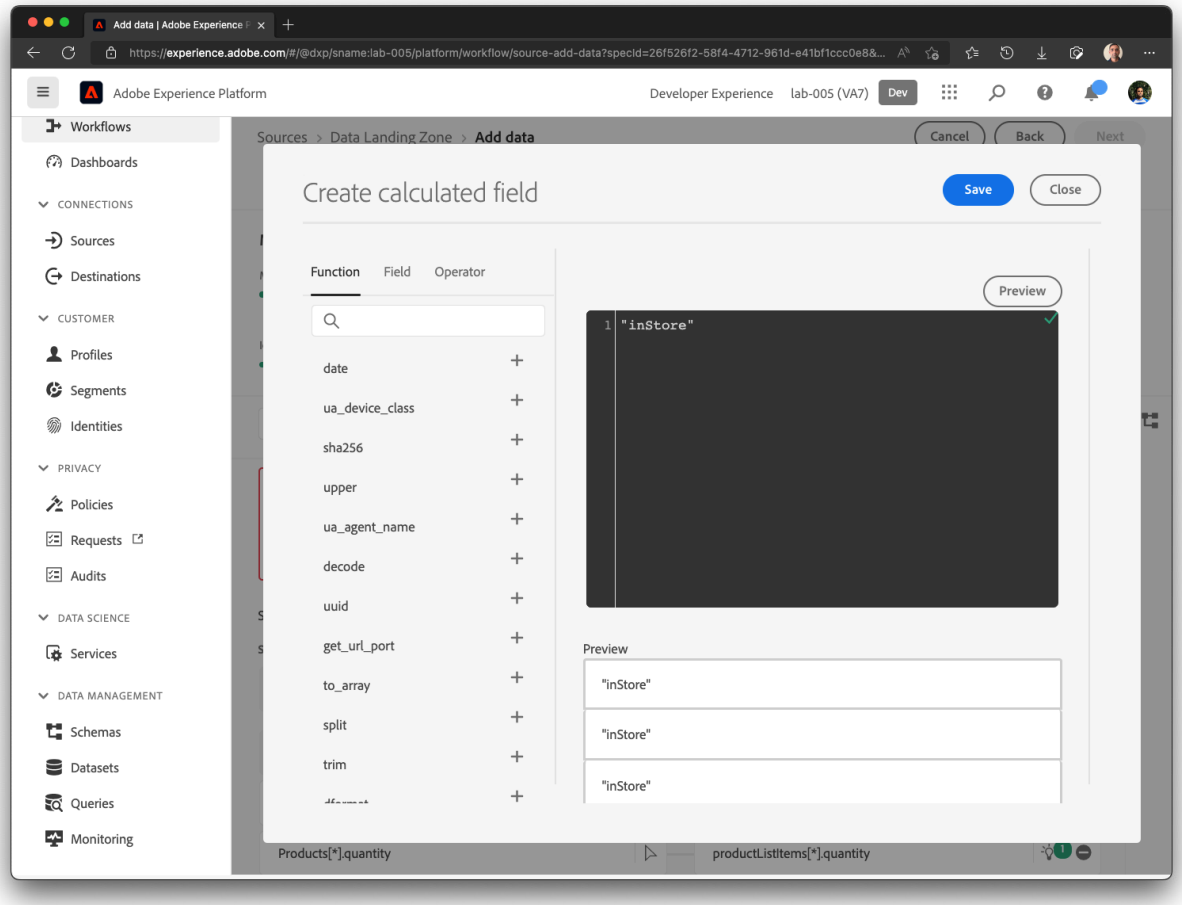
Create the following calculated field by clicking on  and click on Add Calculated Field.



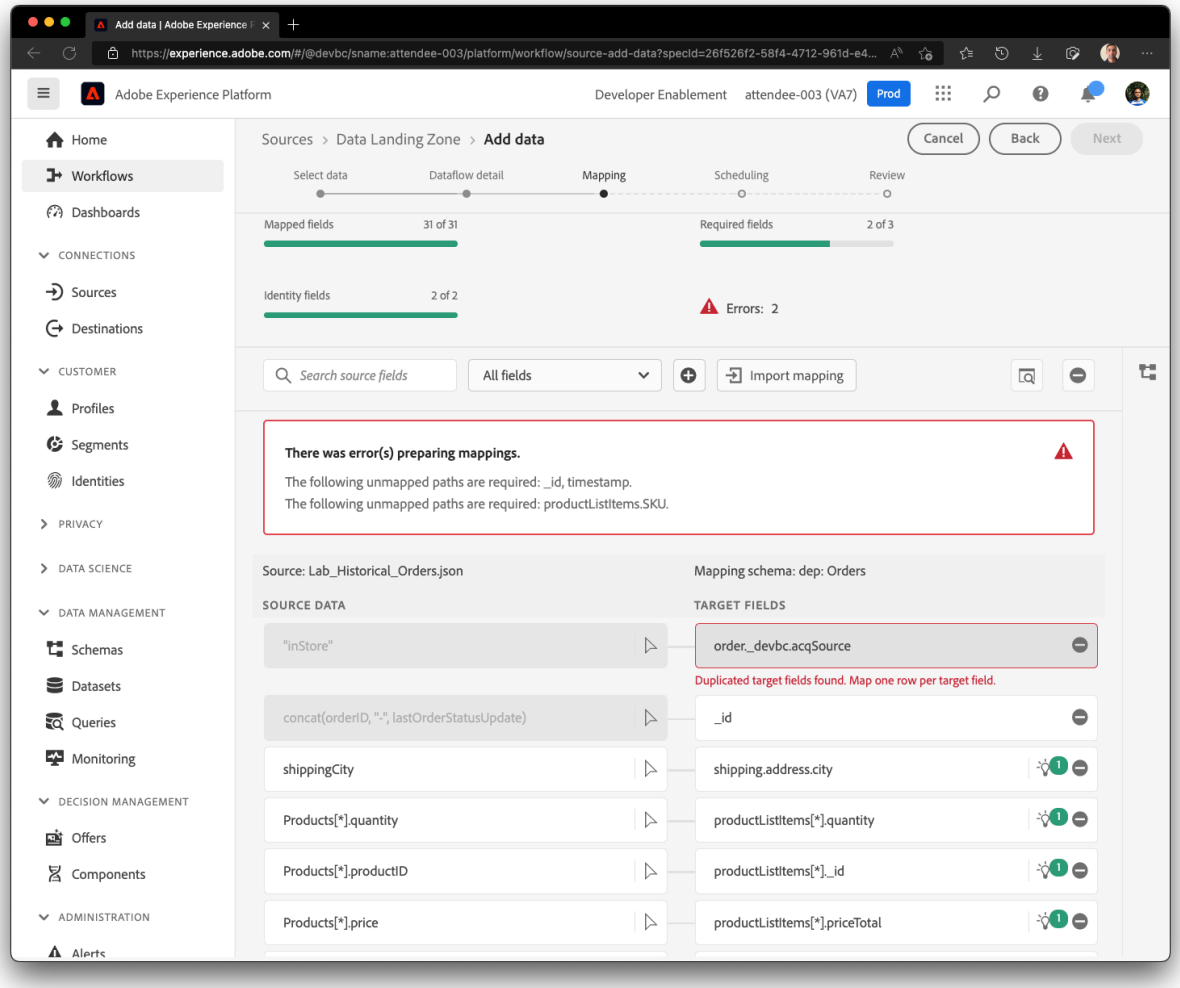
Write the following expression and click Preview. NOTE that this value is case sensitive and must be written exactly this way.

```
"inStore"
```

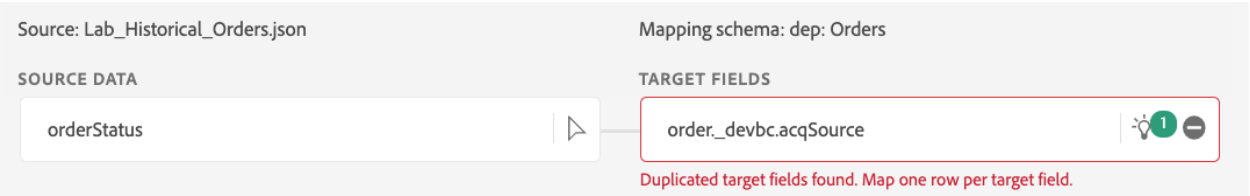
Calculated field will look similar to this. Click Save to save the calculated field.



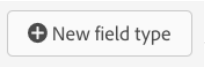
Map the calculated field to **order._devbc.acqSource**. The Data Prep screen will complain there is a duplicate mapping.



Remove the other duplicate mapping for order._devbc.acqSource by clicking on the (-) symbol next to the row.



3.6. Add new pass-through mappings

Add the following passthrough mappings by clicking New field type (

) and Add new field for each row here. Some may already be present due to ML Recommendations.

Source Column	XDM Column
orderStatus	eventType
lastOrderStatusUpdate	timestamp
Products[*]	productListItems[*]
Products[*].productID	productListItems[*].SKU

NOTE that **Products[*].productID** here is mapped to **prodictListItems[*].SKU** in addition to **prodictListItems[*]._id**

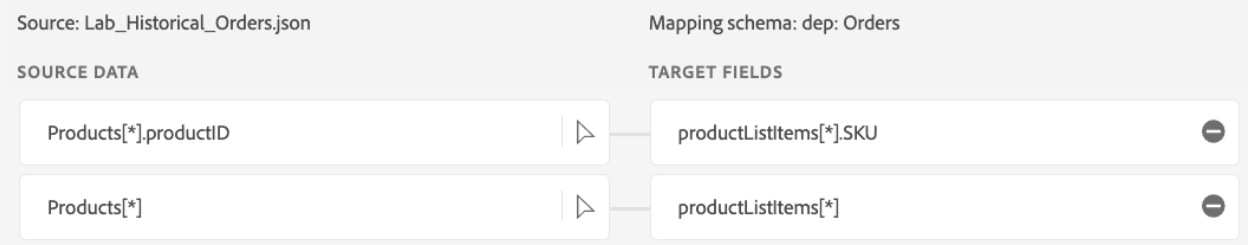
3.7.Product List Items

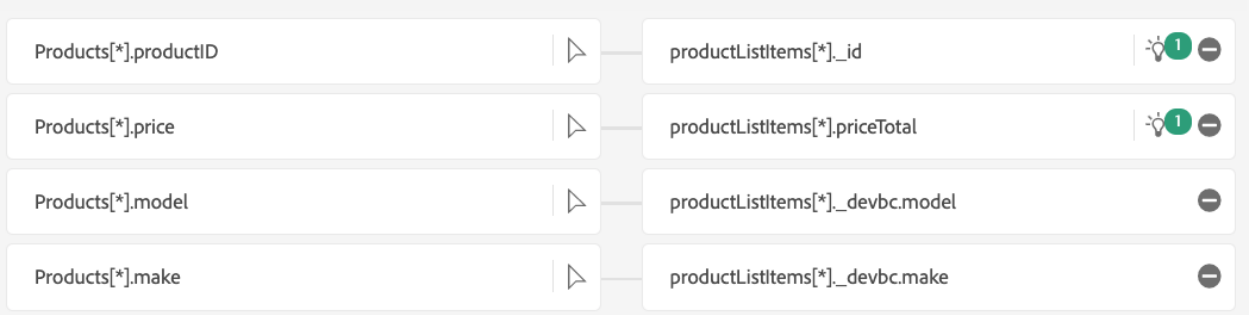
We will remove the mappings to XDM attributes **productListItems[*].quantity** and **productListItems[*].currencyCode**. Change the other mappings to **productListItems[*]**

NOTE	You will have to manually type in [*] in the target attributes after making a selection for **Products** source object array or **productListItems** XDM attribute.

Source Column	XDM Column	Action
products[*]	productListItems[*]	Add
products[*].productID	productListItems[*].SKU	Add
products[*].productID	productListItems[*]._id	No change
products[*].make	productListItems[*]._devbc.make	Change
products[*].model	productListItems[*]._devbc.model	Change
products[*].price	productListItems[*].priceTotal	No change
products[*].quantity	productListItems[*].quantity	Remove
products[*].currencyCode	productListItems[*].currencyCode	Remove

The resultant mappings for **ProductListItems[*]** should look like this.





3.8. Final Mapping Set

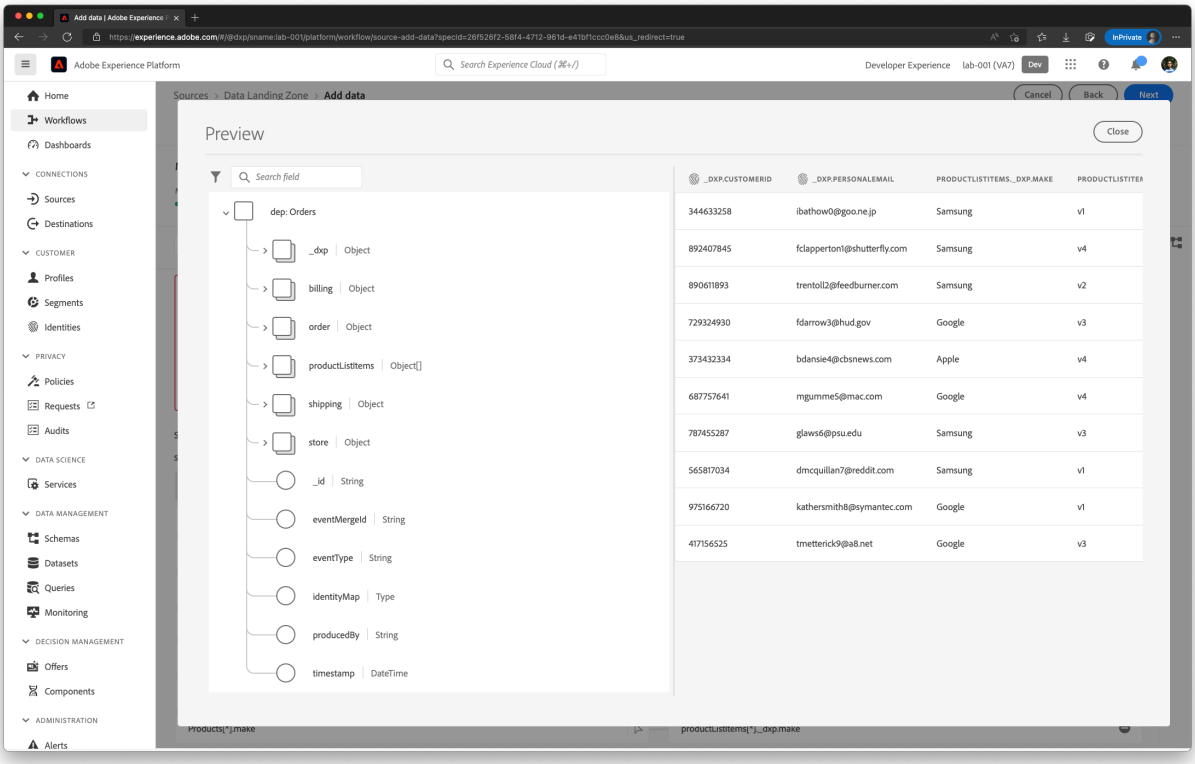
The final mapping set should look like this:

#	Source Column	XDM Column	✓
1	orderStatus	eventType	
2	lastOrderStatusUpdate	timestamp	
3	orderID	order.orderID	
4	orderDate	order.orderDate	
5	orderTotal	order.priceTotal	
6	paymentType	order.payment.paymentType	
7	paymentAmount	order.payment.paymentAmount	
8	paymentCurrencyCode	order.payment.currencyCode	
9	paymentTransactionID	order.payment.transactionID	
10	plan.ID	order._devbc.plan.planID	
11	customerID	_devbc.customerID	
12	personalEmail	_devbc.personalEmail	
13	storeID	store.storeID	
14	shippingStreetAddress	shipping.address.street1	
15	shippingCity	shipping.address.city	
16	shippingState	shipping.address.state	
17	shippingZip	shipping.address.postalCode	
18	shippingMethod	shipping.shippingMethod	
19	shippingAmount	shipping.shippingAmount	
20	shippingDestination	shipping.shippingDestination	
21	billingStreetAddress	billing.address.street1	
22	billingCity	billing.address.city	

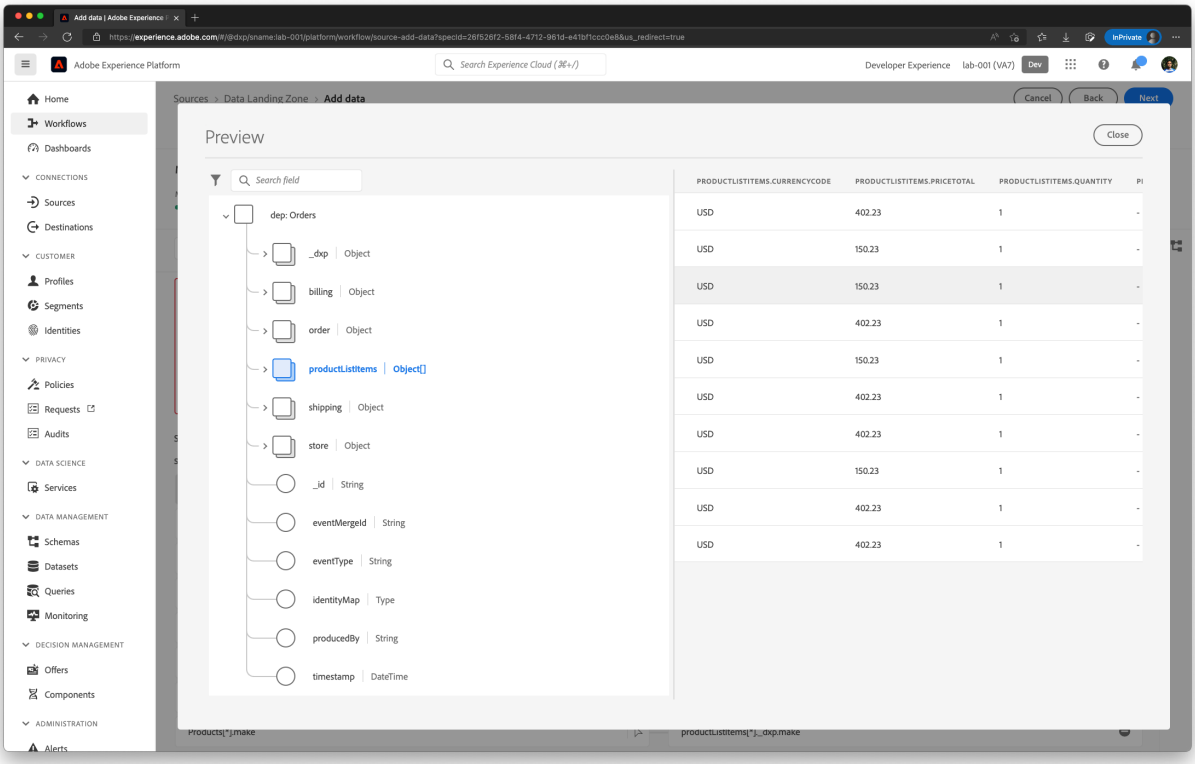
#	Source Column	XDM Column	✓
23	billingState	billing.address.state	
24	billingZip	billing.address.postalCode	
25	products[*]	productListItems[*]	
26	products[*].productID	• productListItems[*]._id • productListItems[*].SKU	
27	products[*].make	productListItems[*]._devbc.make	
28	products[*].model	productListItems[*]._devbc.model	
29	products[*].price	productListItems[*].priceTotal	
	Calculated Fields		
30	concat(orderID, "-", lastOrderStatusUpdate)	_id	
31	"inStore"	order._devbc.acqSource	

3.9. Preview the data

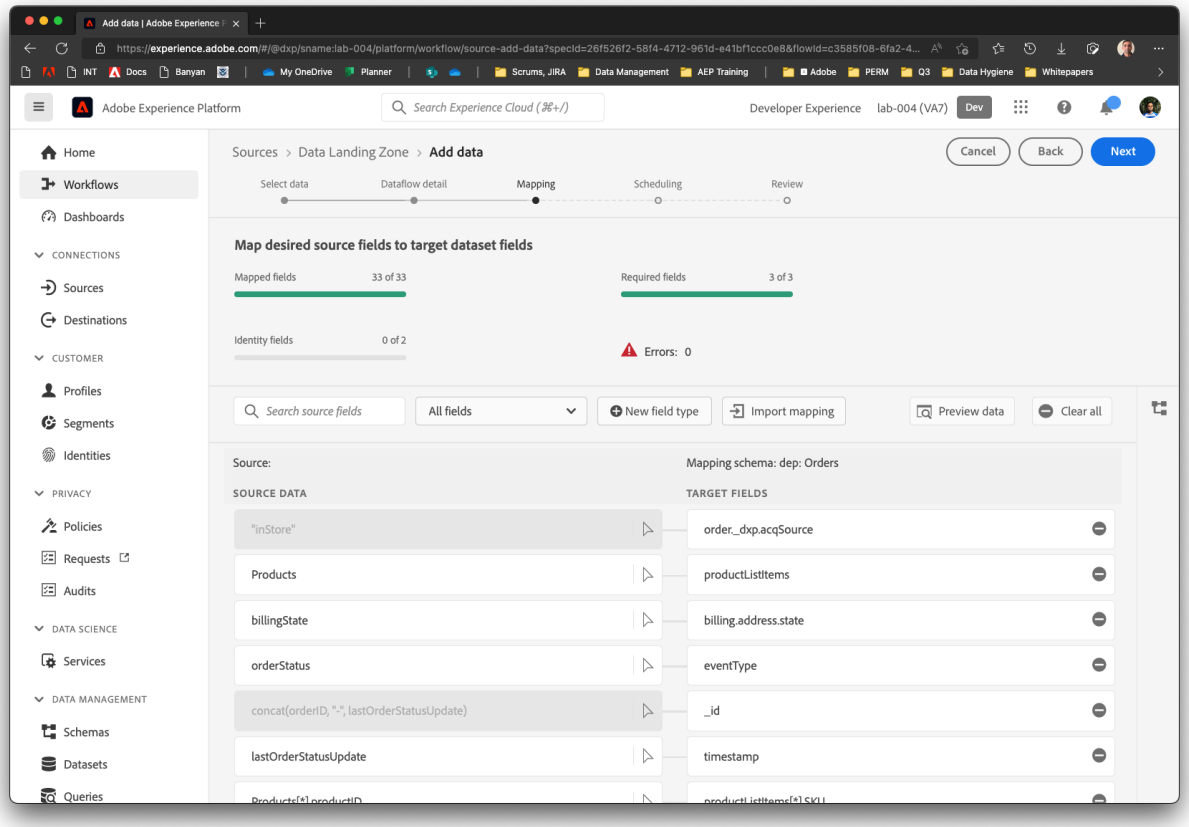
Preview the mapping output. Scroll through all the attributes to ensure there is no red exclamation next to any of the attributes on the right hand side. Preview will look similar to the below screenshot



On the left hand side navigation of the Preview, select **productListItems** object array. Right hand side will update to show only the attributes in that object array. Notice that **productListitems.currencyCode** and **productListitems.quantity** are automatically populated (even after removing the mappings). This happens because **productListItems** as a parent object is mapped.

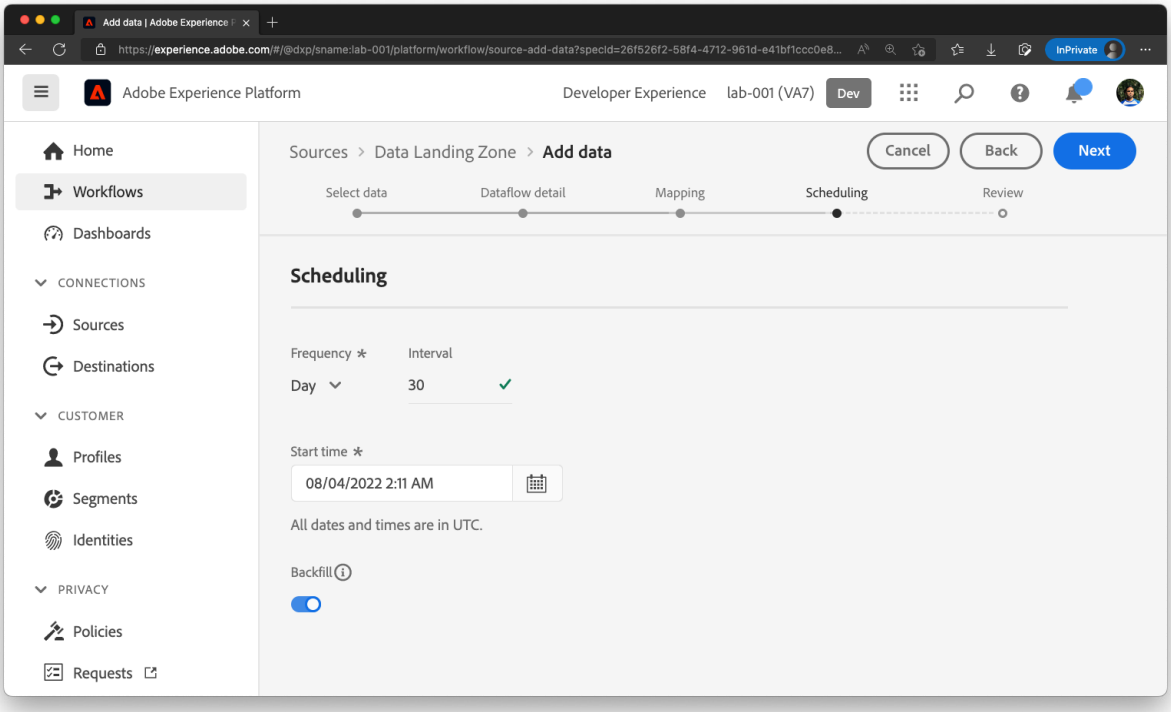


Completed mapping will look similar to the following screenshot

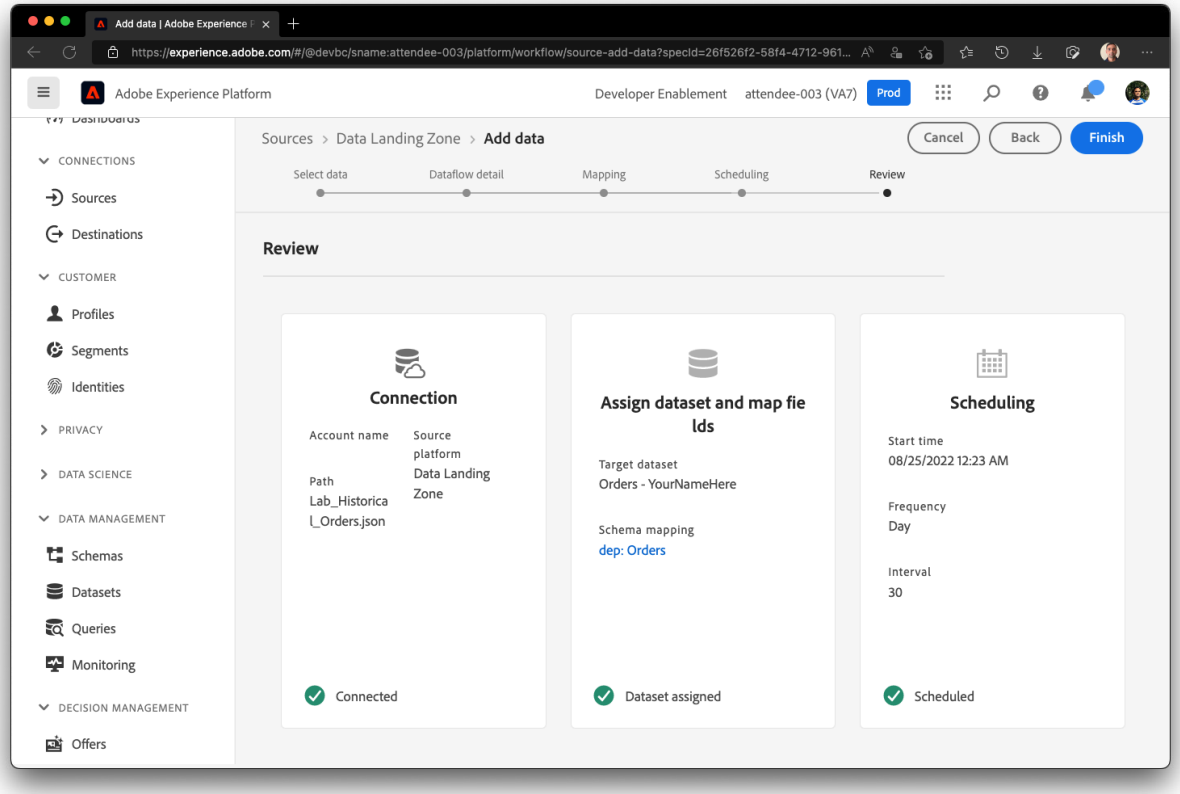


3.10. Schedule

Set the schedule to run every 30 days by setting the Frequency as **Day** and Interval as **30**.



Review the flow and click Finish



Backfill / Historical load Dataflow will be created. Dataflow execution will not start immediately and will take few minutes. So, last Dataflow Run Status will be set to "*No runs*".

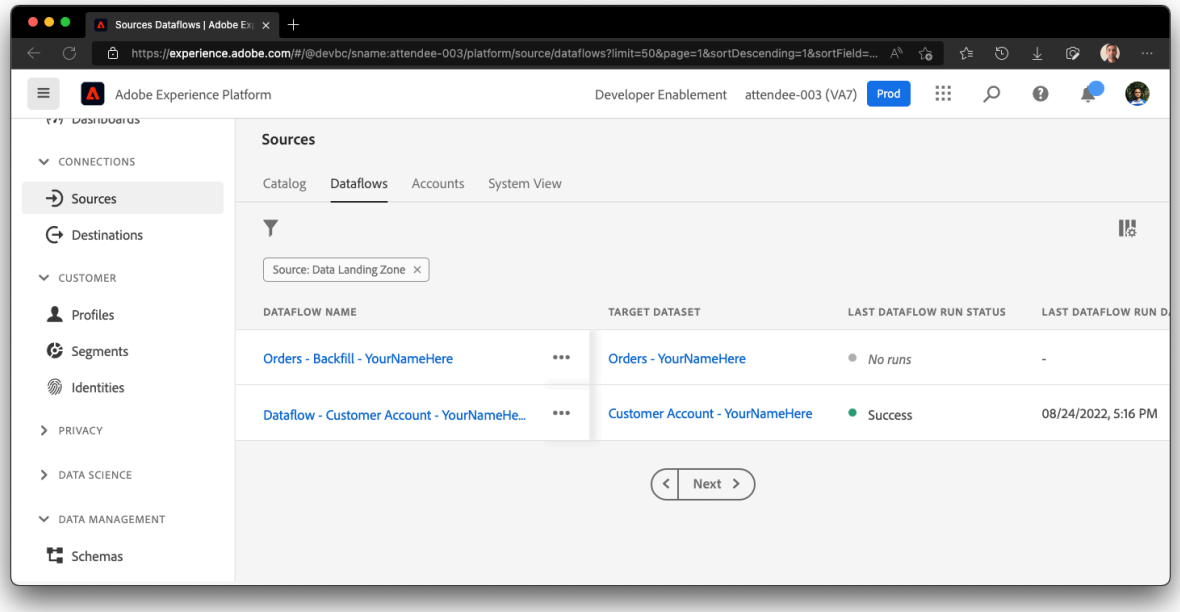
Adobe Experience Platform

Lab 4 - Data Pipeline Streaming

Adobe Experience Platform Bootcamp Deep Dive Edition

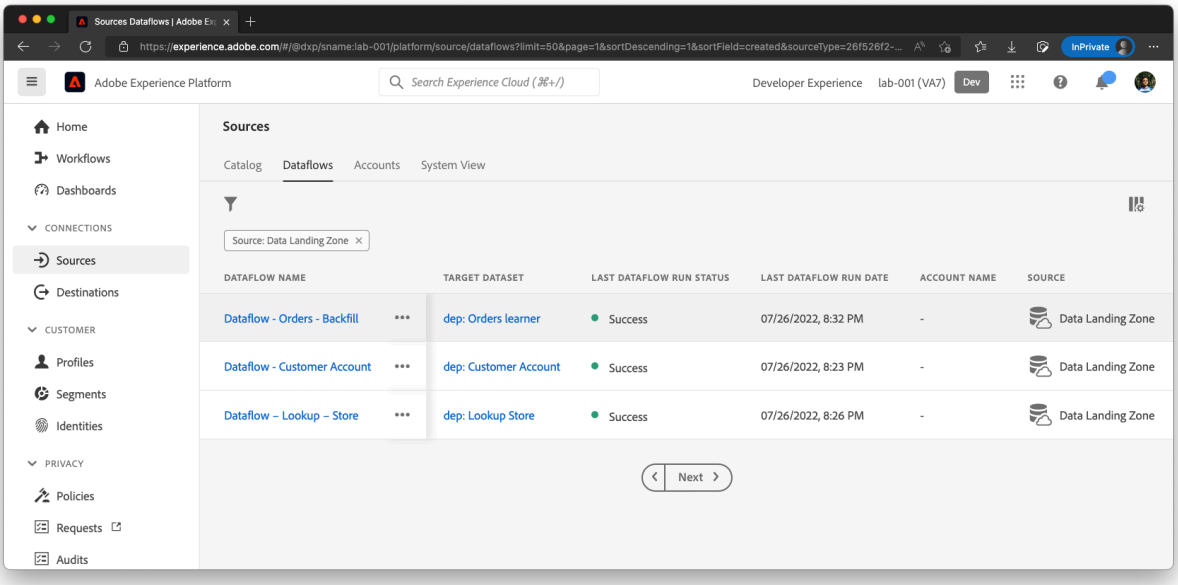
Name _____

Sandbox _____

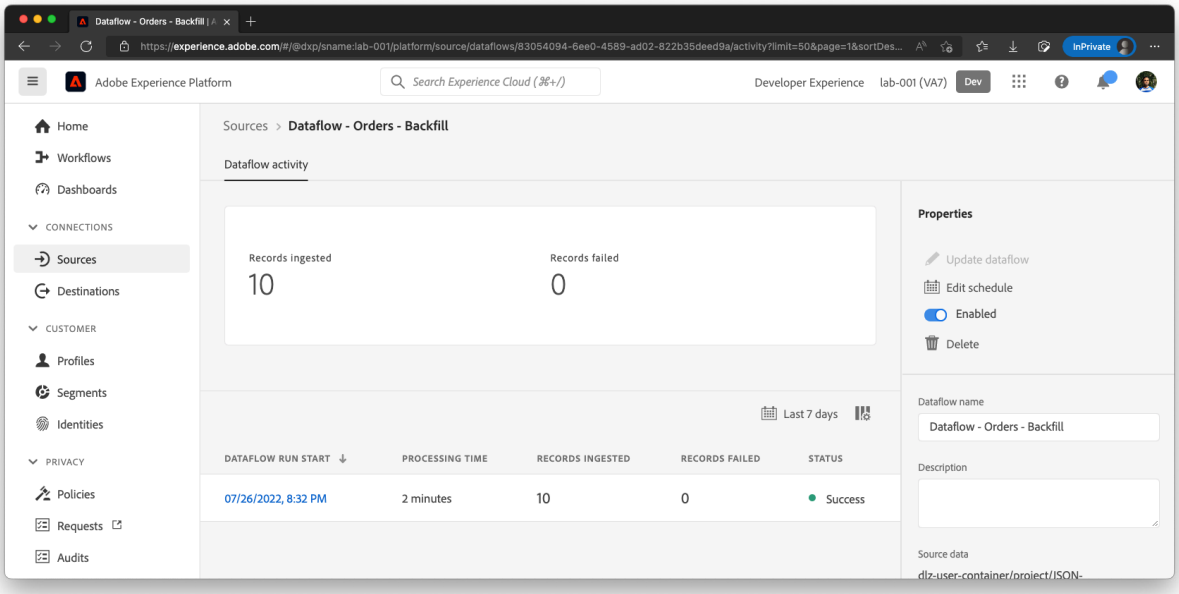


3.11. Check scheduled execution

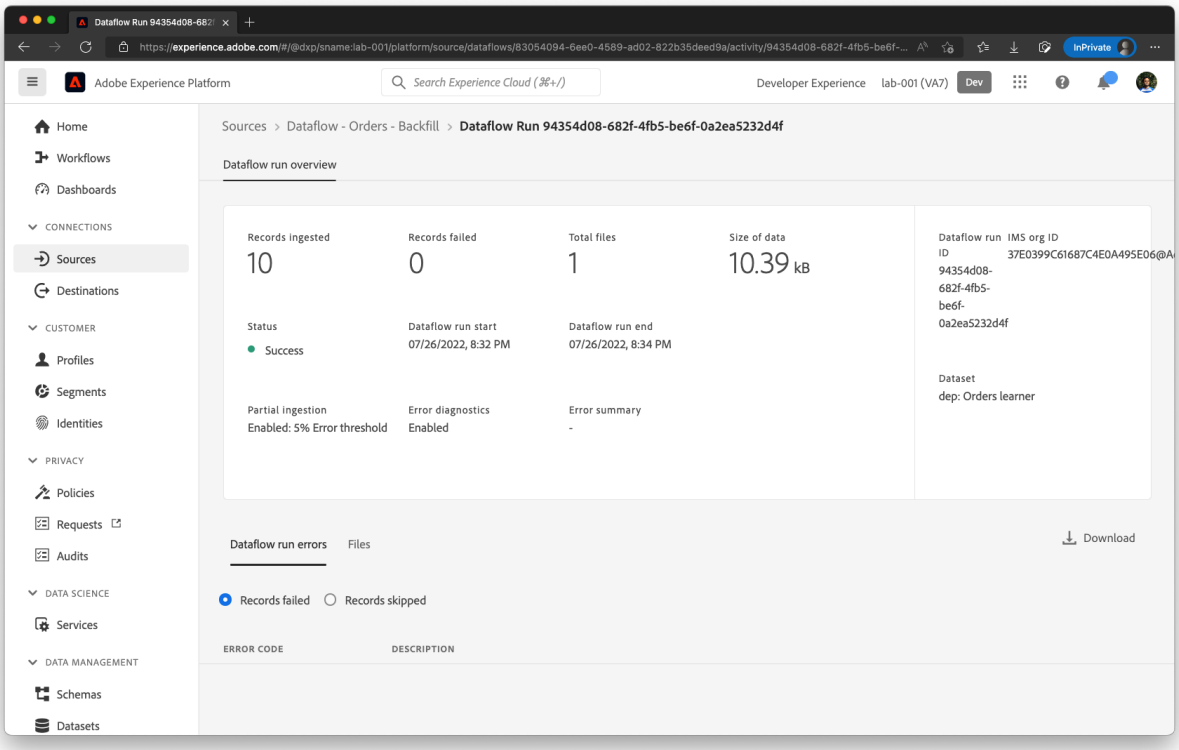
After few minutes, the Dataflow will succeed. Notice the **Last Dataflow Run Status** and **Last Dataflow Run Date**.



Click on the Dataflow name to get a list of Dataflow Runs. 10 Records should be ingested.

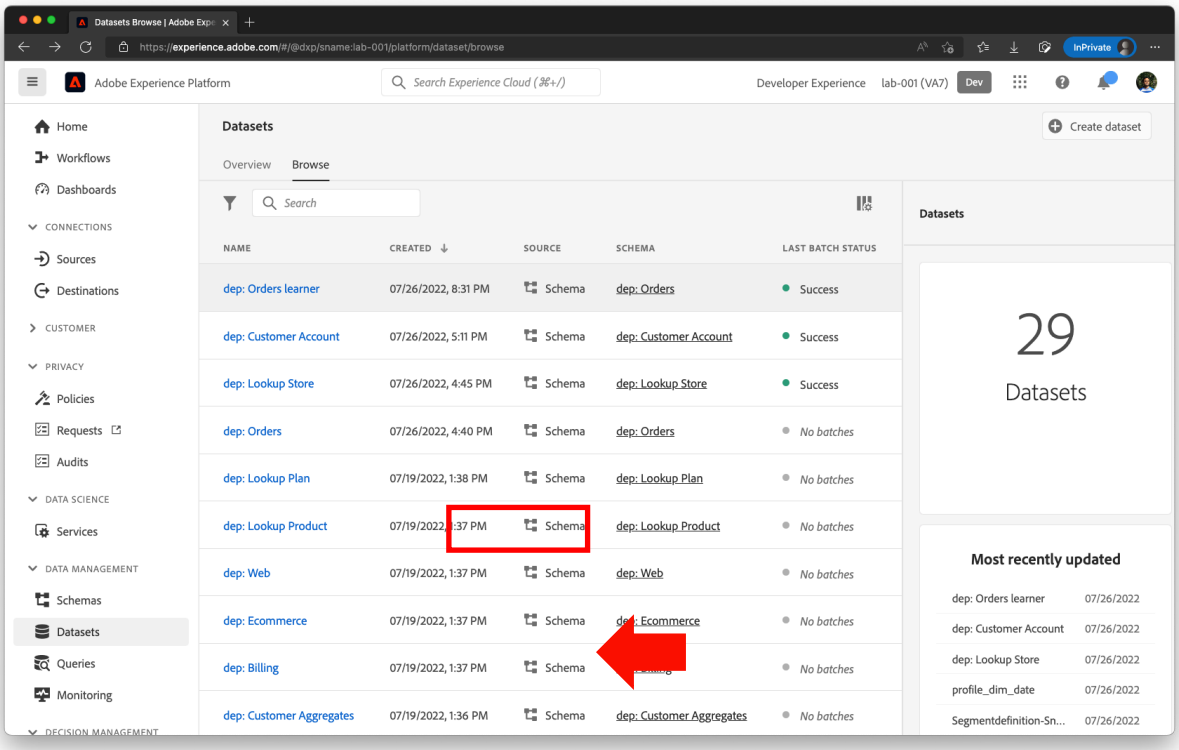


Click on the Dataflow Run Start time to see error diagnostic details.

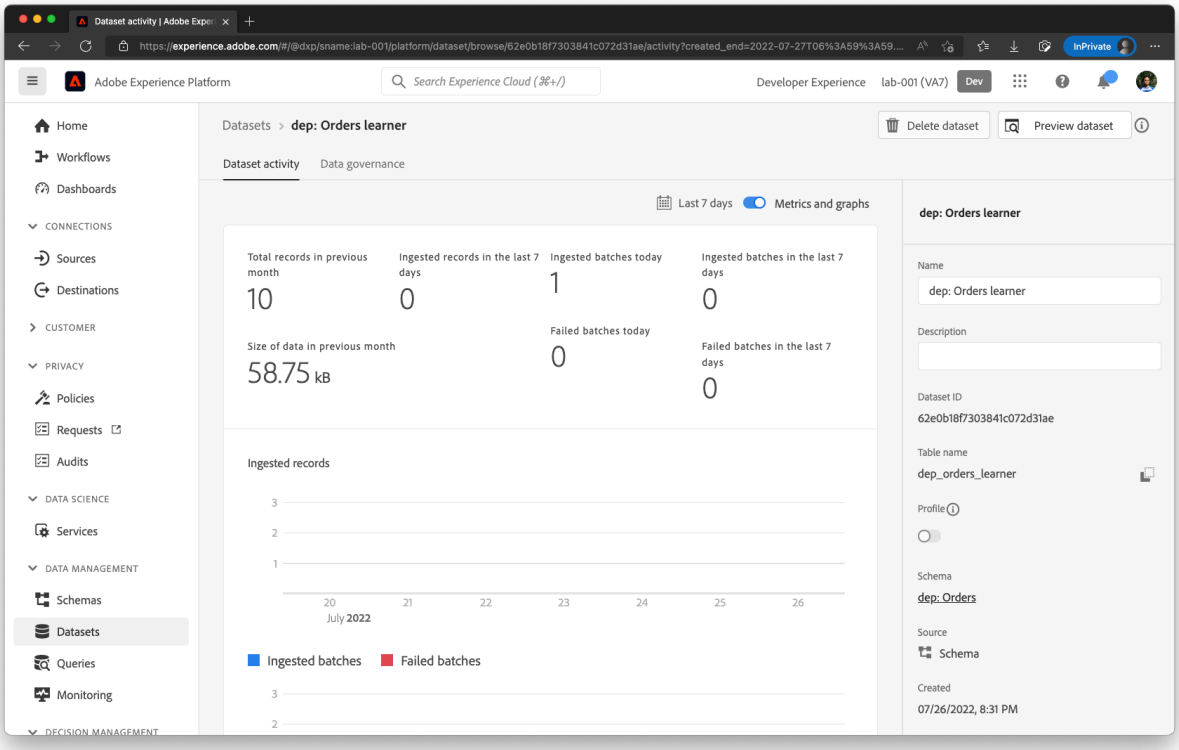


3.12. Verify the data

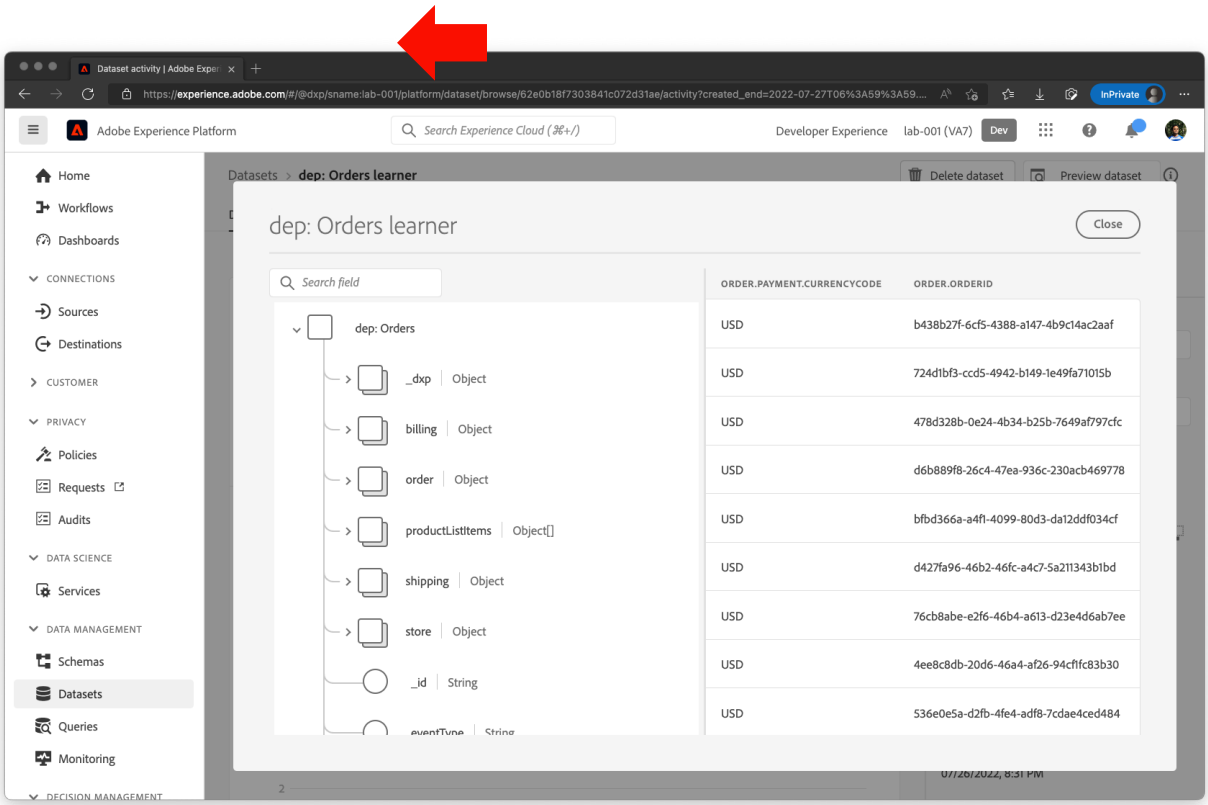
In the Left Nav bar, Go to Datasets in Platform and click on **Orders - YourNameHere**



Click on the **Preview Dataset.**



Notice that **productListitems.currencyCode** and **productListItems.quantity** are auto populated.



1. Lab Overview

In this lab, we will use Import Mapping capability to import Customer Accounts and Orders historical data mapping into streaming ingestion flow. Data will be ingested using Streaming HTTP API.

Expected time: 30 minutes

2. Learning Objectives

What should you walk away with after taking this Lab?

- Creating a Streaming Inlet
- Importing mapping set from another Dataflow
- Get Dataflow ID, and Dataset ID from the UI
- Using REST API to ingest an event

3. Lab Tasks A – Customer Accounts – Live data (Streaming)

In this exercise, we will load the Customer Account data from Streaming source to AEP Data Lake and Profile.

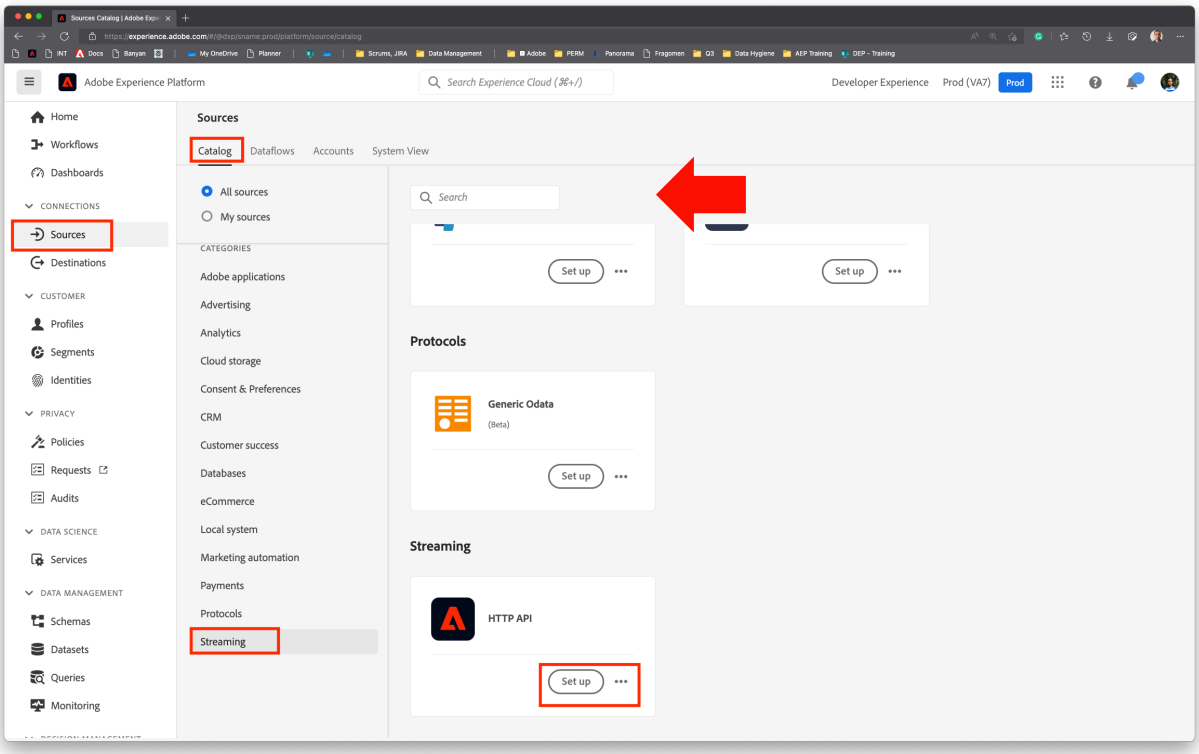
Pre-requisites

1. Lab 2 is successfully completed

Steps

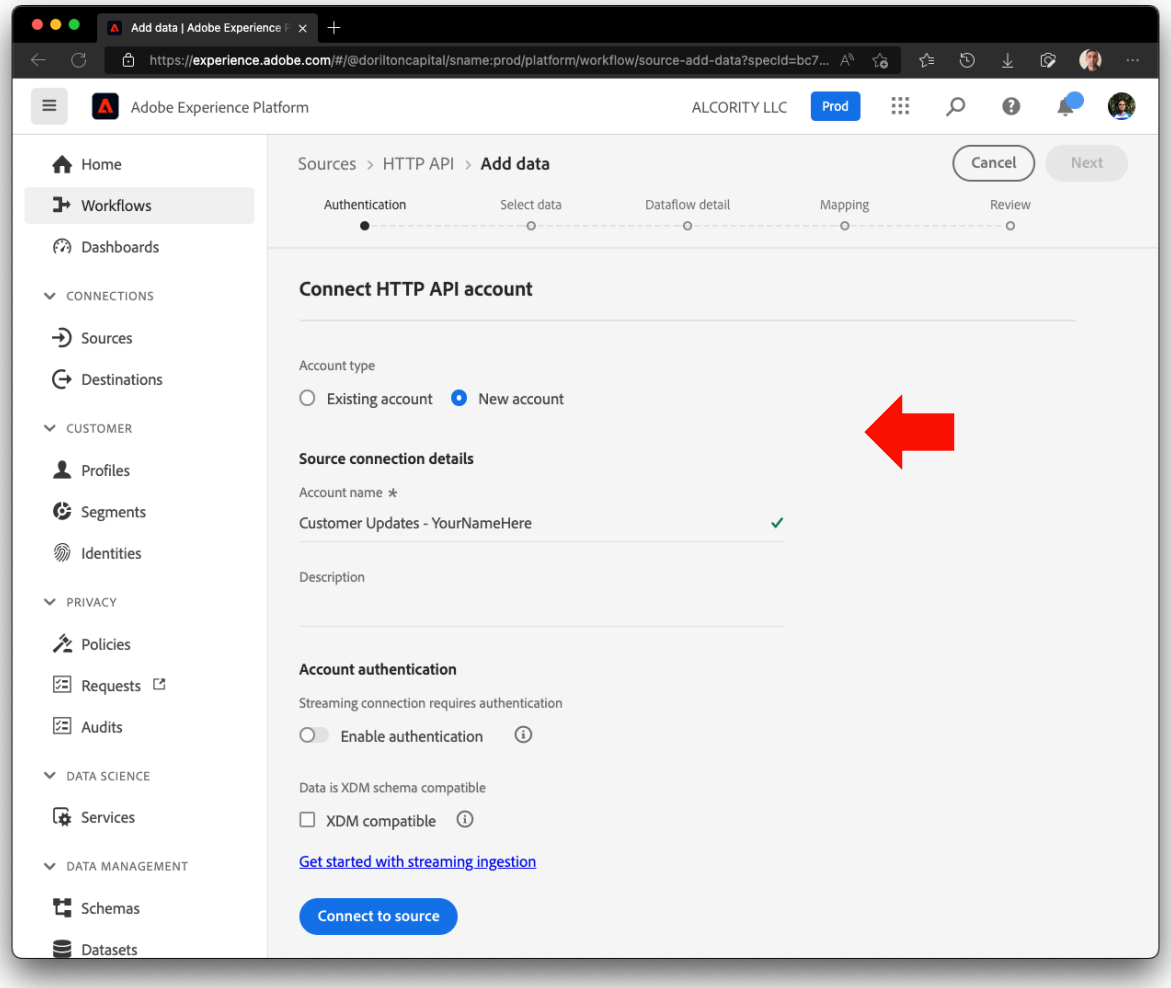
Go to Adobe Experience Platform → **Sources** → **Catalog** → **Streaming**. Click on **Setup** / **Add Data** for the HTTP API.

Tip	If at least one connection exists for that source, you will see "**Add data**" as the default action. If no connections exist for that source, you will see "**Setup**" as the default action.

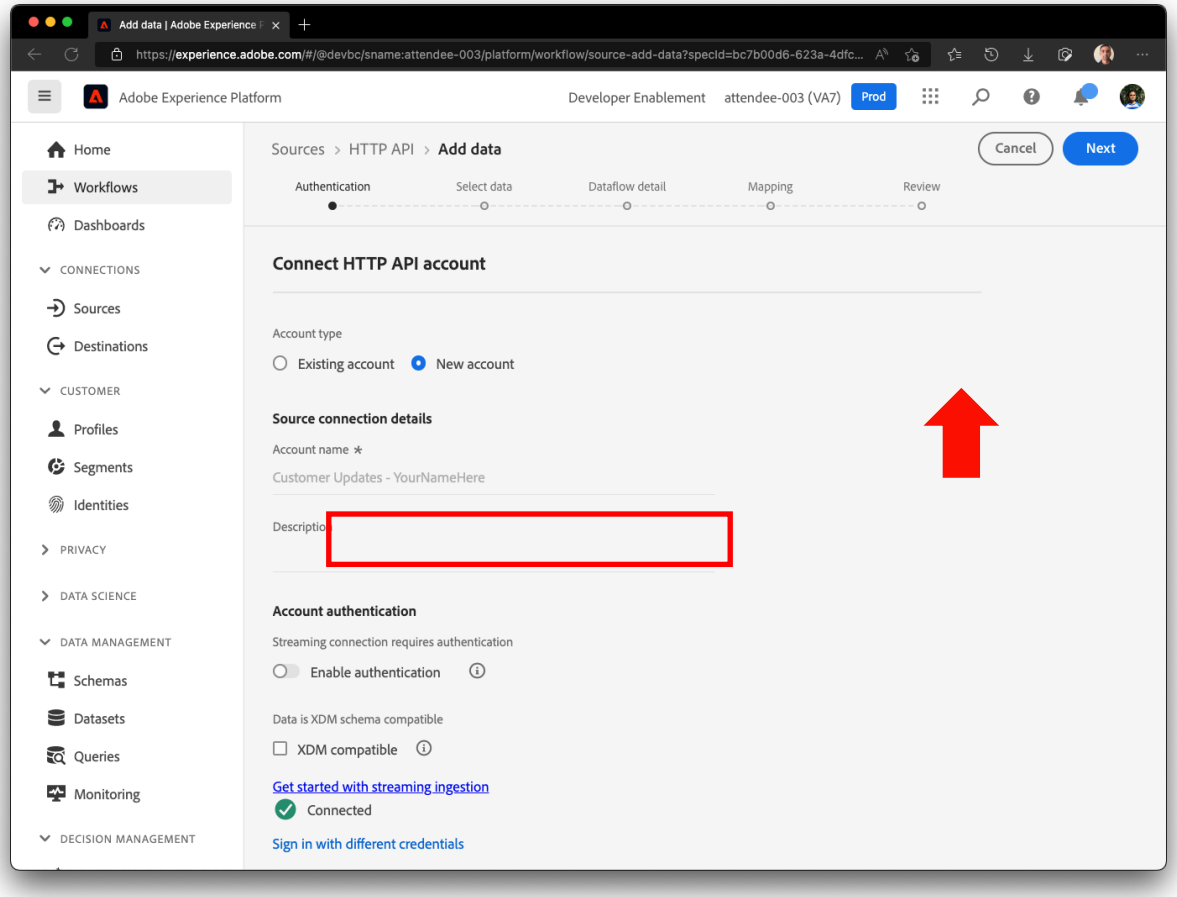


3.1. Creating a Streaming Inlet

In the Connect HTTP API Account screen, Choose **New account** and name the account as **Customer Updates – YourNameHere**. Enable authentication is turned OFF by default. XDM compatible checkbox is OFF too. We will leave these defaults as-is. Then click **Connect to source**.

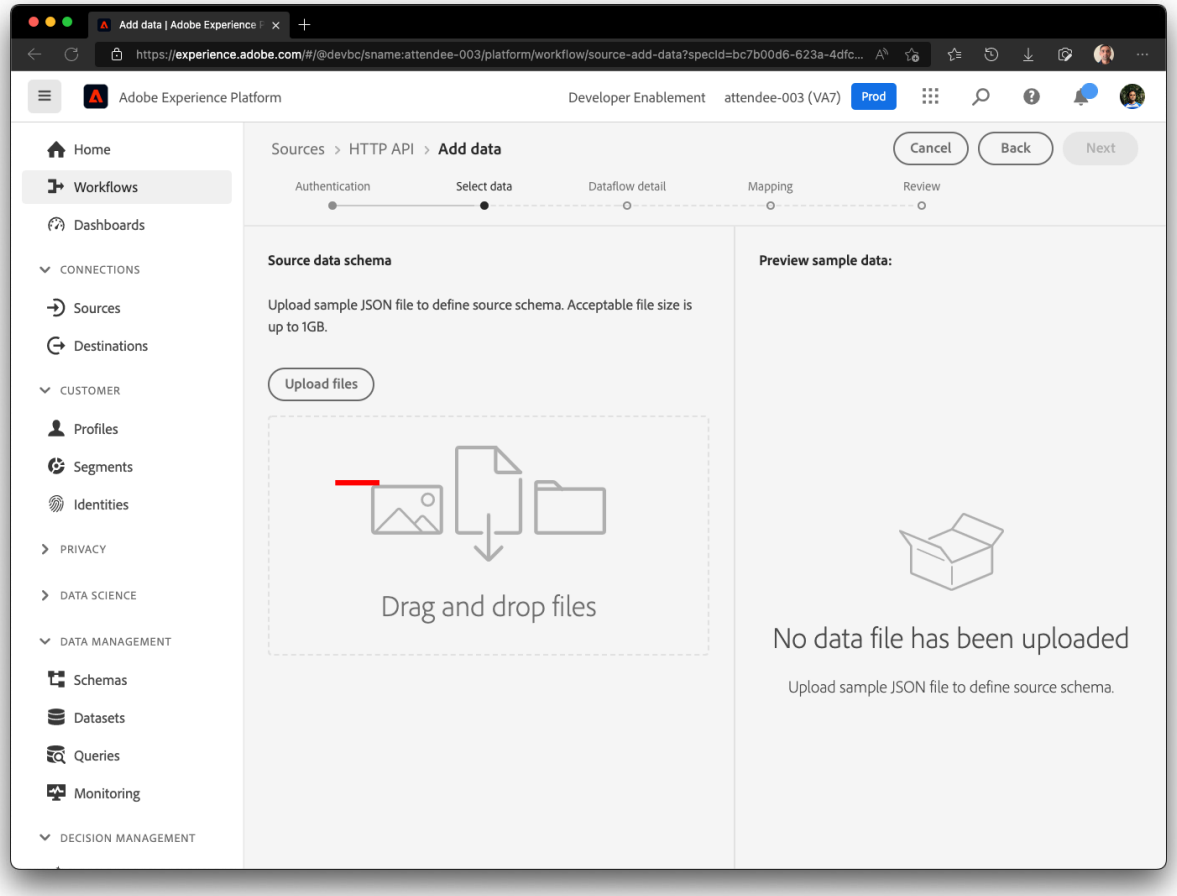


A Streaming account / endpoint is created for you, and you will see message called "**Connected**" on the screen. Click next to continue.

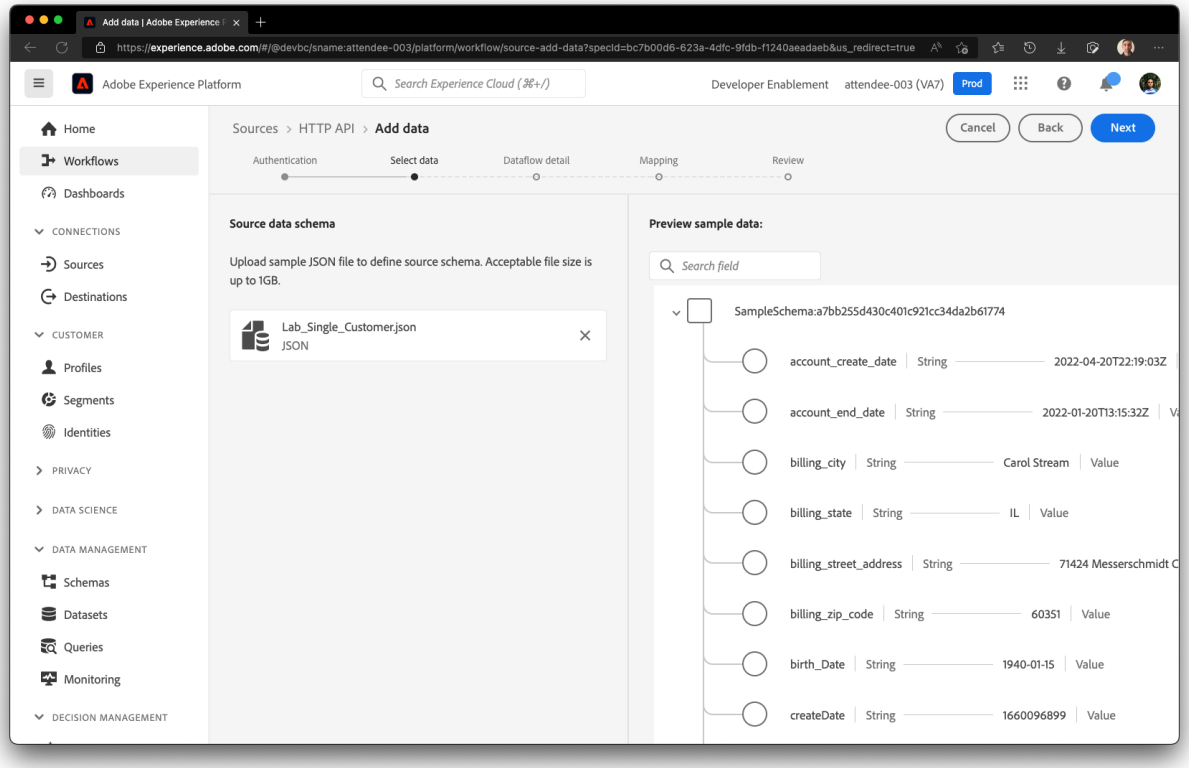


3.2. Upload sample data

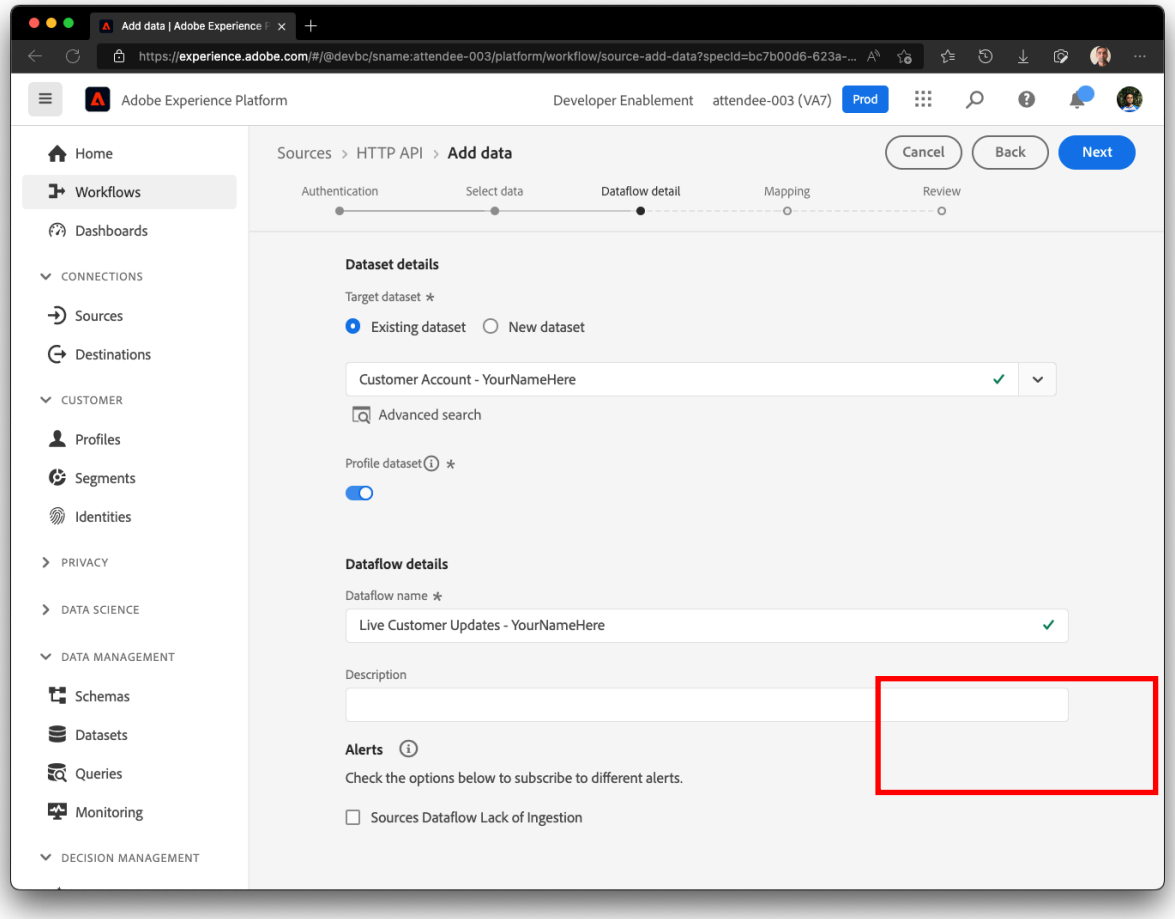
In the Source data schema screen, upload the JSON file **Lab_Single_Customer.json** from your local file system to the AEP



Once the file is uploaded, preview will appear as follows. Click Next to continue.

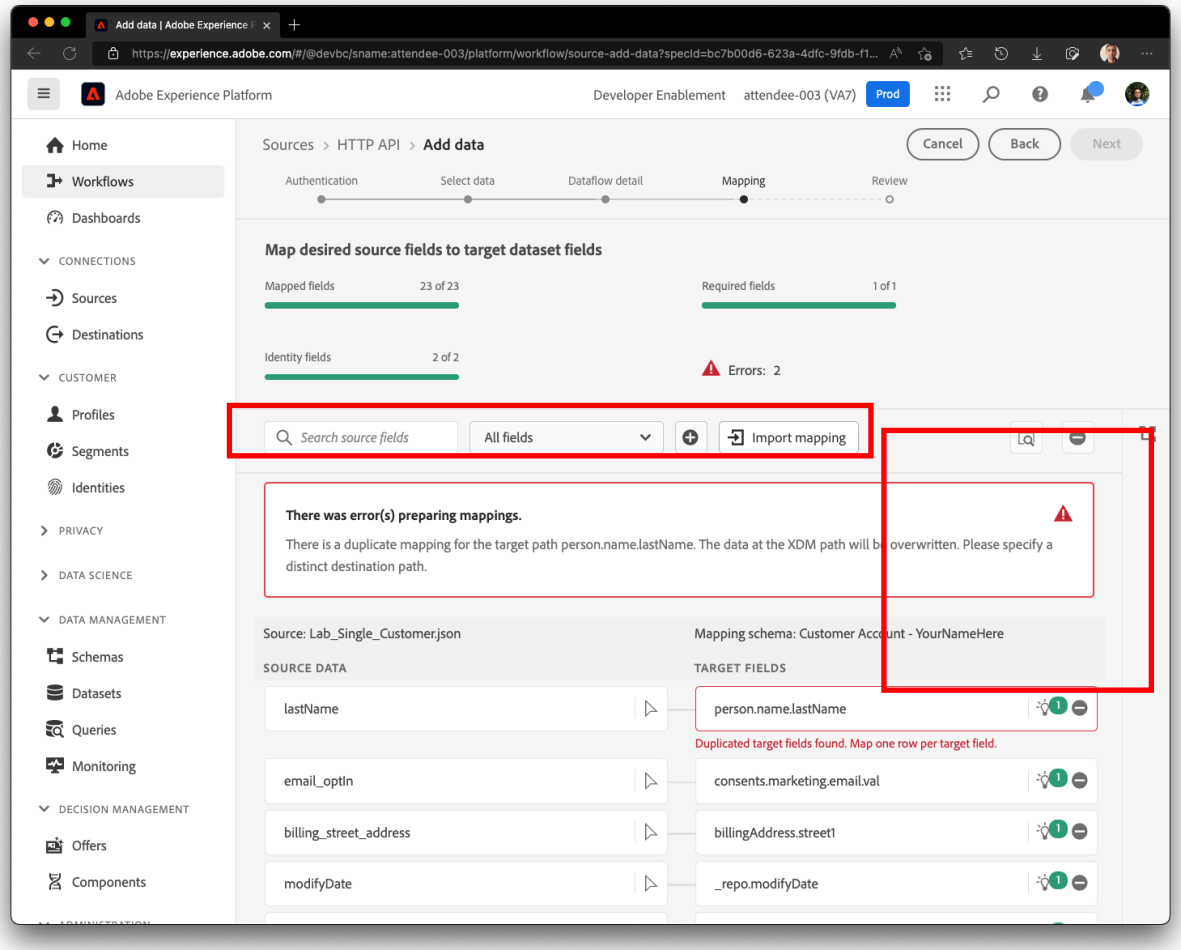


In the Dataset details screen, choose **Existing dataset** and select the dataset you created in Lab 2. It was named **Customer Account – YourNameHere**. Profile dataset is toggled ON by default. Name the Dataflow as Live **Customer Updates – YourNameHere**.



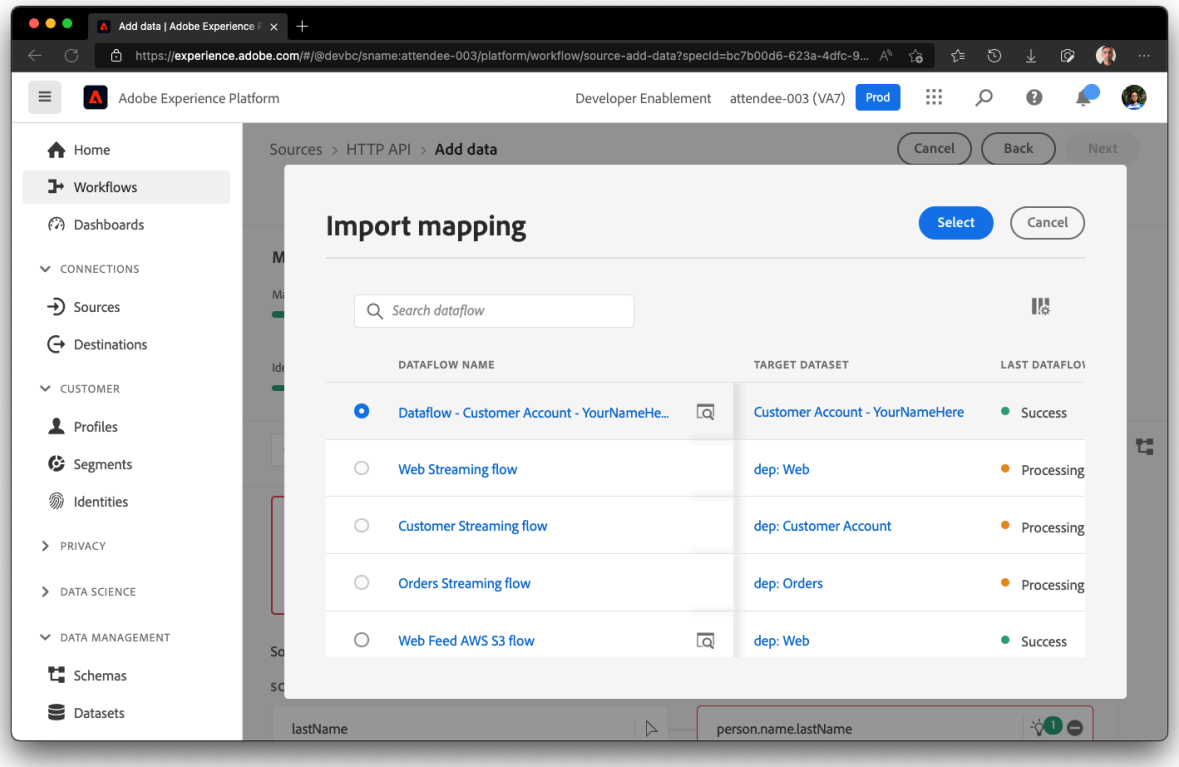
3.3. Import the Customer Accounts Mapping

Mapping screen will load. ML recommendations will pre-populate some attributes. Some errors may also appear. Click on the Import Mapping button (Import mapping) to import the mapping we created in Lab 2 for batch customer accounts ingestion.

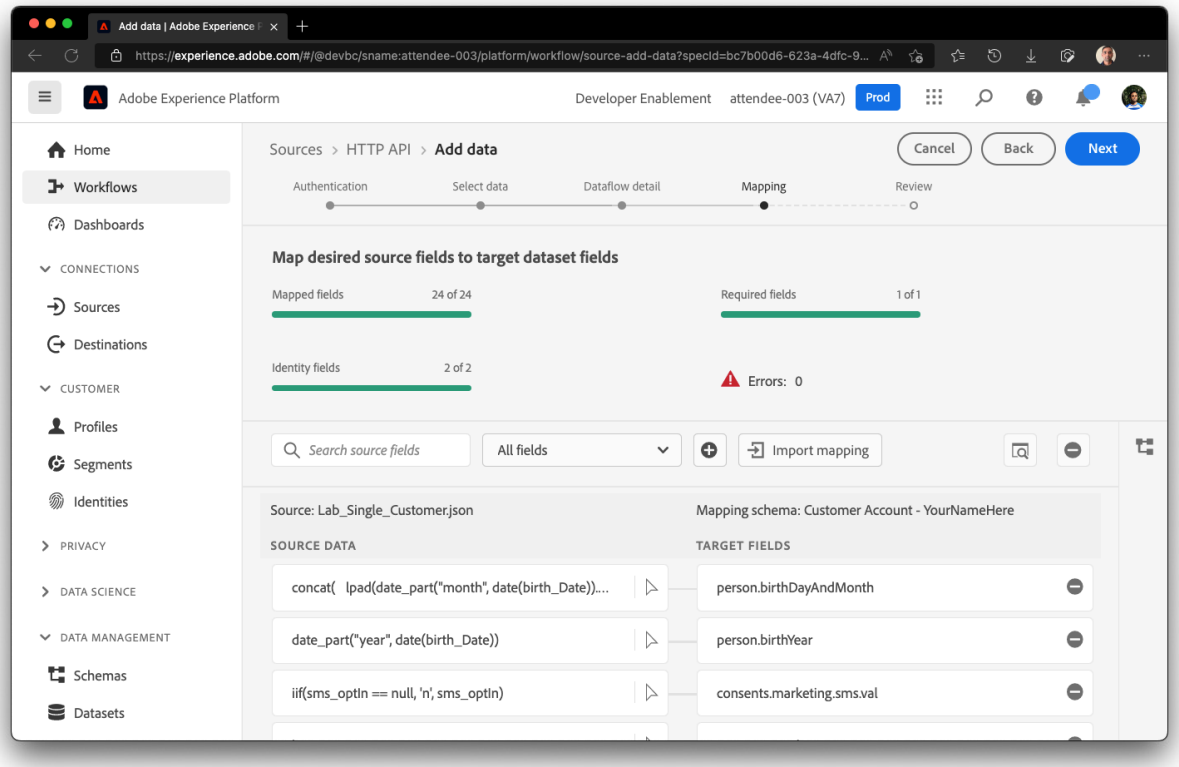


Choose the Dataflow created in the Lab 2. It was named **Dataflow – Customer Account – YourNameHere**.

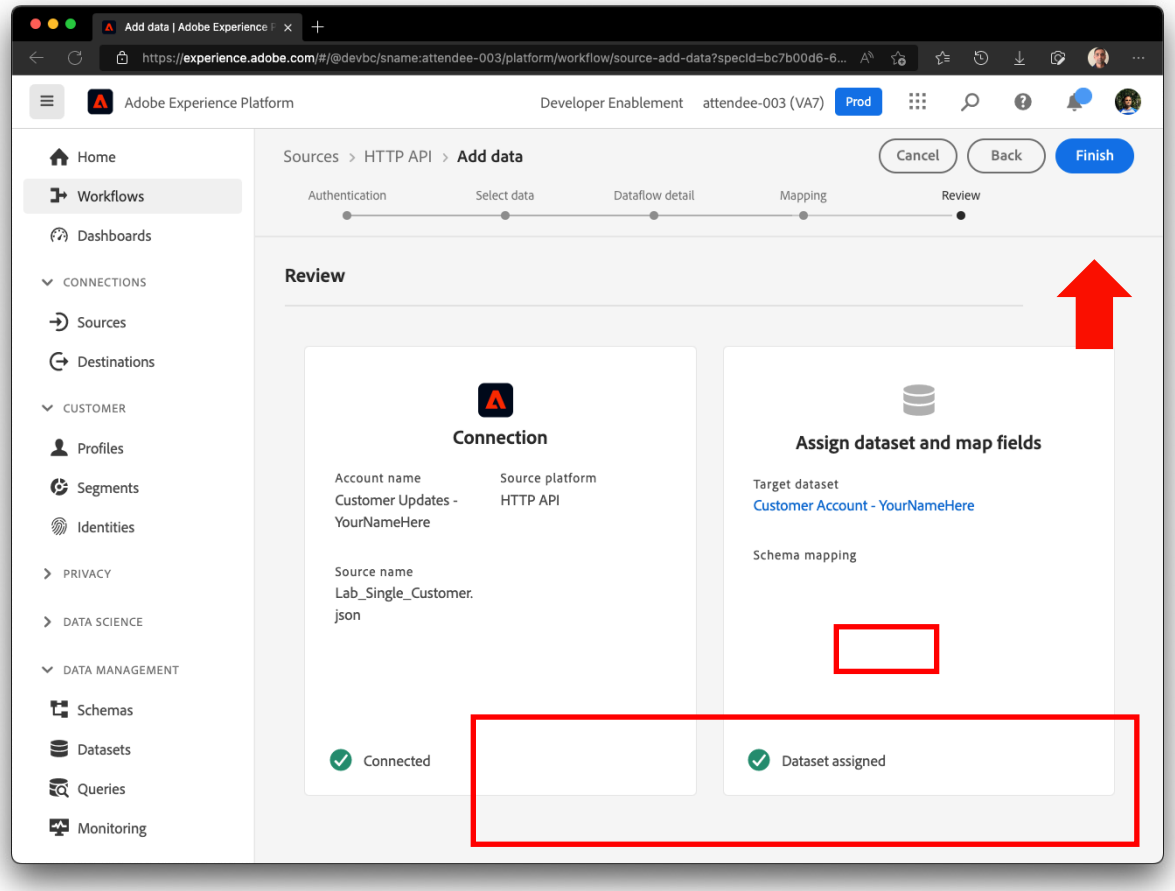
Warning	If you have not completed the Lab 2 successfully, you will NOT be able to proceed further. Additionally, if you have not updated the Lab 2 dataflow to change the transformation logic for **person.birthDayAndMonth** XDM attribute, this lab will fail.



Mapping is now imported. Ensure the calculated field mapped to **person.birthDayAndMonth** has "lpad" string in it. Click next to continue.



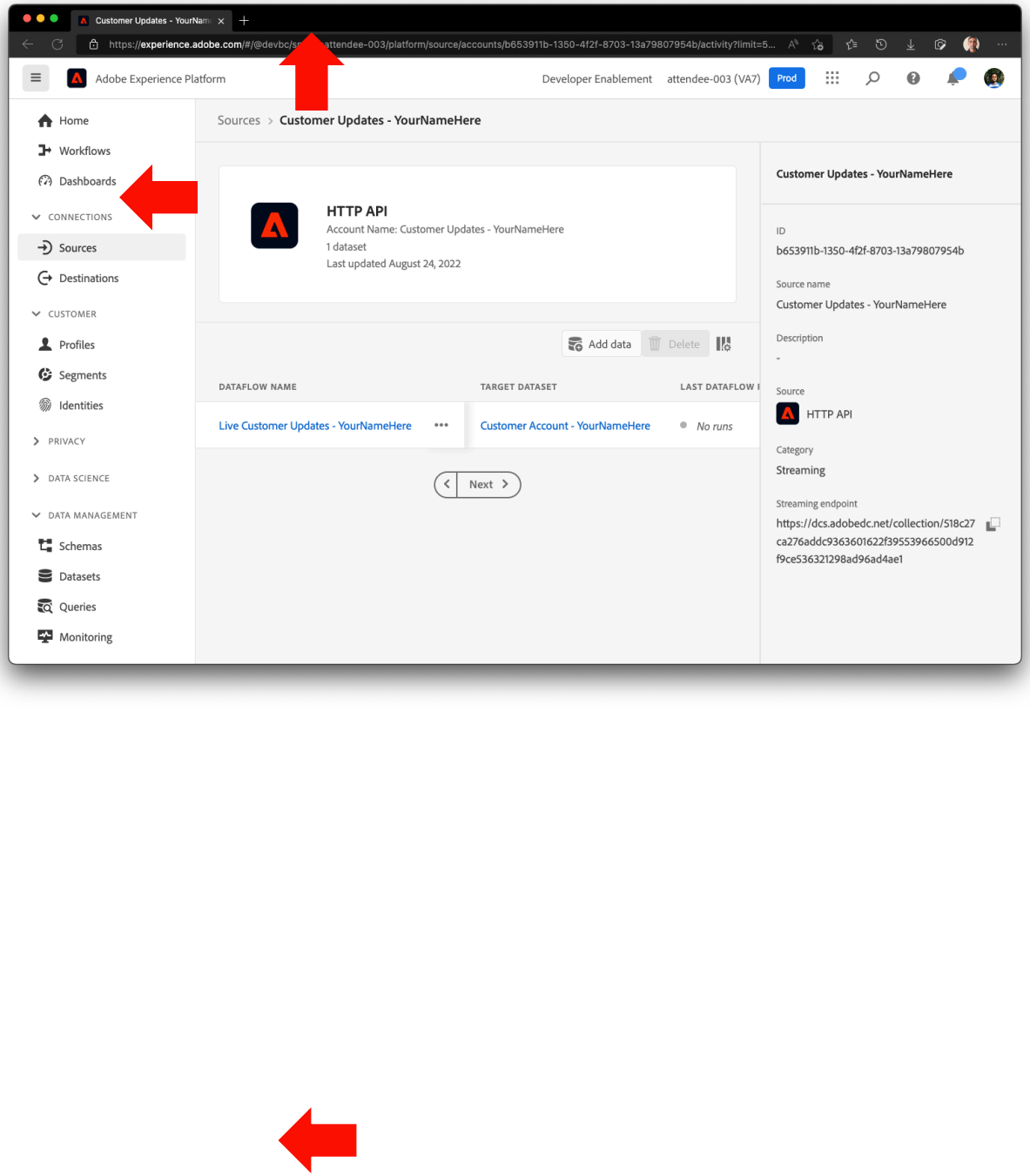
Review the details and click Finish to save the Dataflow.



3.4. Streaming Endpoint

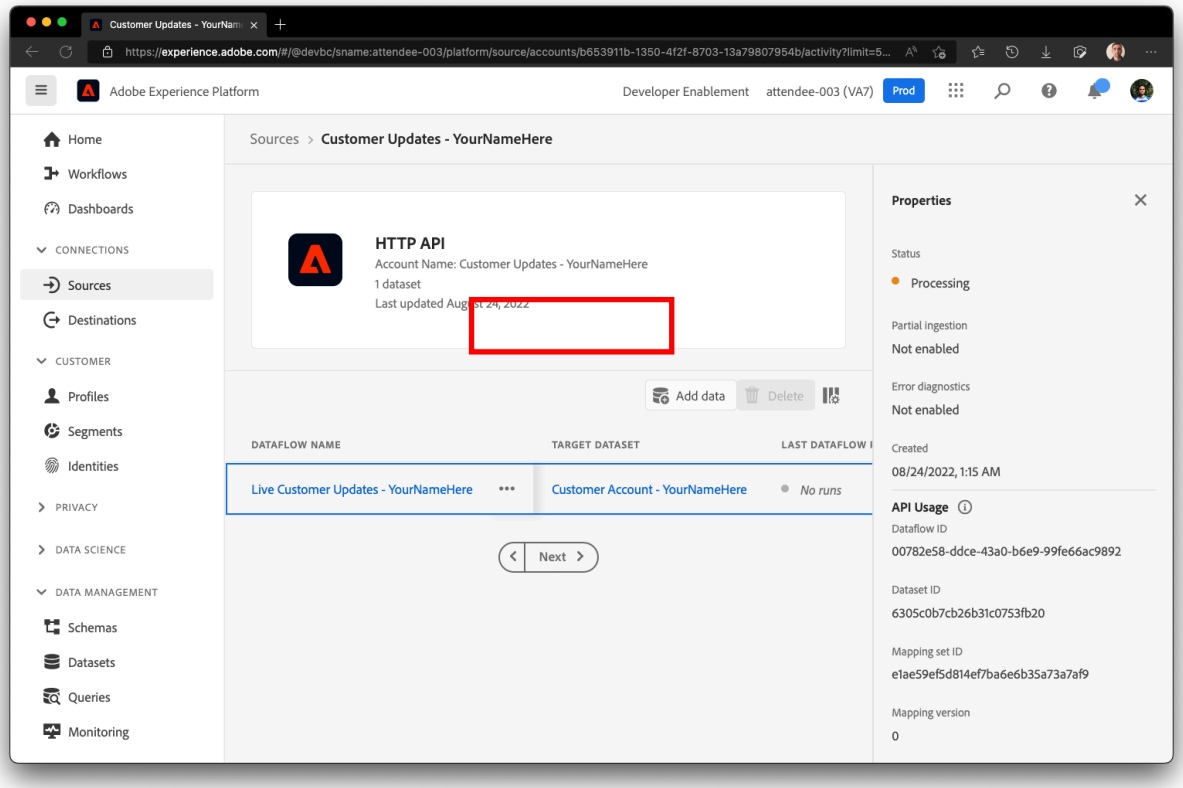
In the screen that appears, gather the streaming endpoint, we will use it in the API.

Tip	If you do not see this information, most likely the Dataflow is selected on the left. Click on the Dataflow row (not the blue link) on the left again to unselect it.



Get Dataflow ID and Dataset ID

In the current screen, select the dataflow name, scroll down the right hand panel and take a note of the Dataflow ID and Dataset ID from the **API Usage** section.



3.5. Place a REST API call

Switch to the Postman UI, and set the following

Type: **POST**

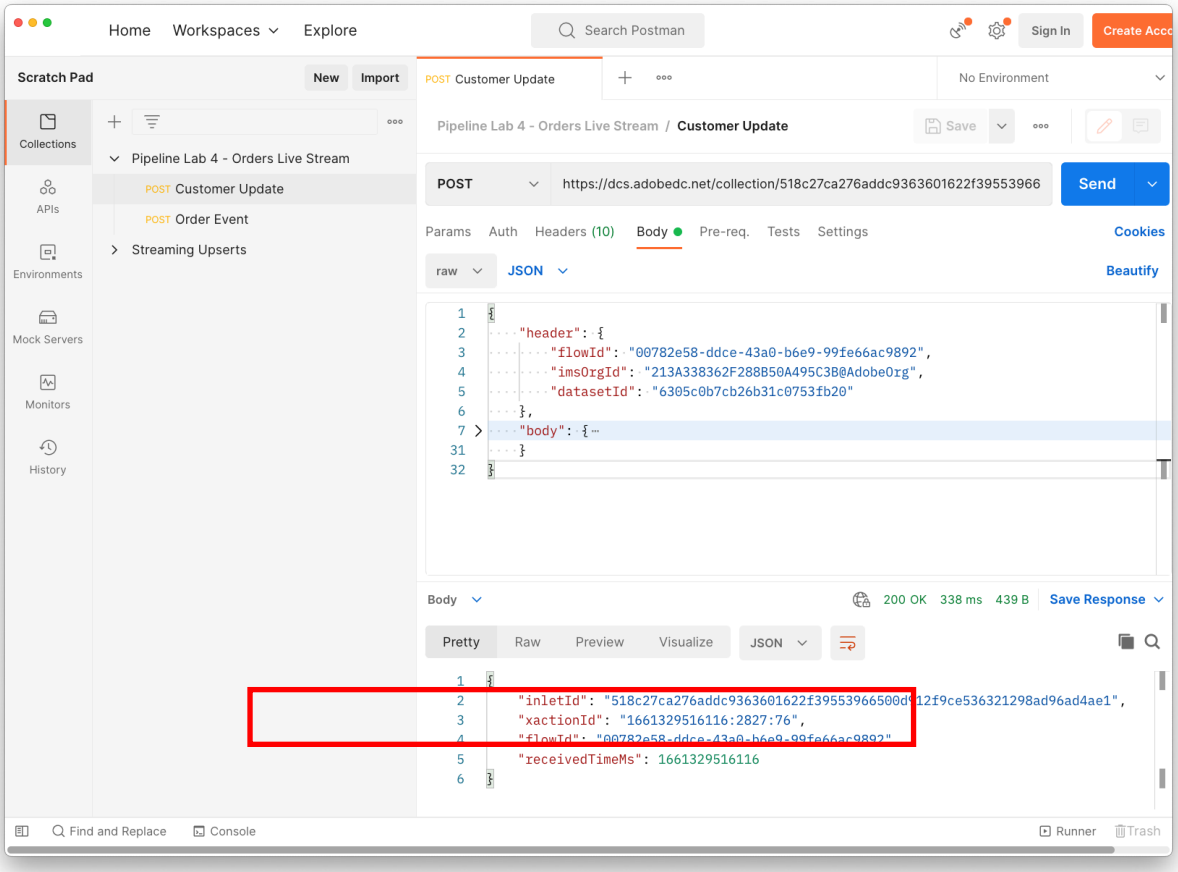
URL: « Streaming Endpoint » you captured in the previous step

Body: The below structure.

Replace your flowId, and datasetId.

```
{
  "header": {
        "flowId": "00782e58-ddce-43a0-b6e9-99fe66ac9892",
        "imsOrgId": "213A338362F288B50A495C3B@AdobeOrg",
        "datasetId": "6305c0b7cb26b31c0753fb20"
    },
  "body": {

  }
}
```



Add the JSON request body as follows.

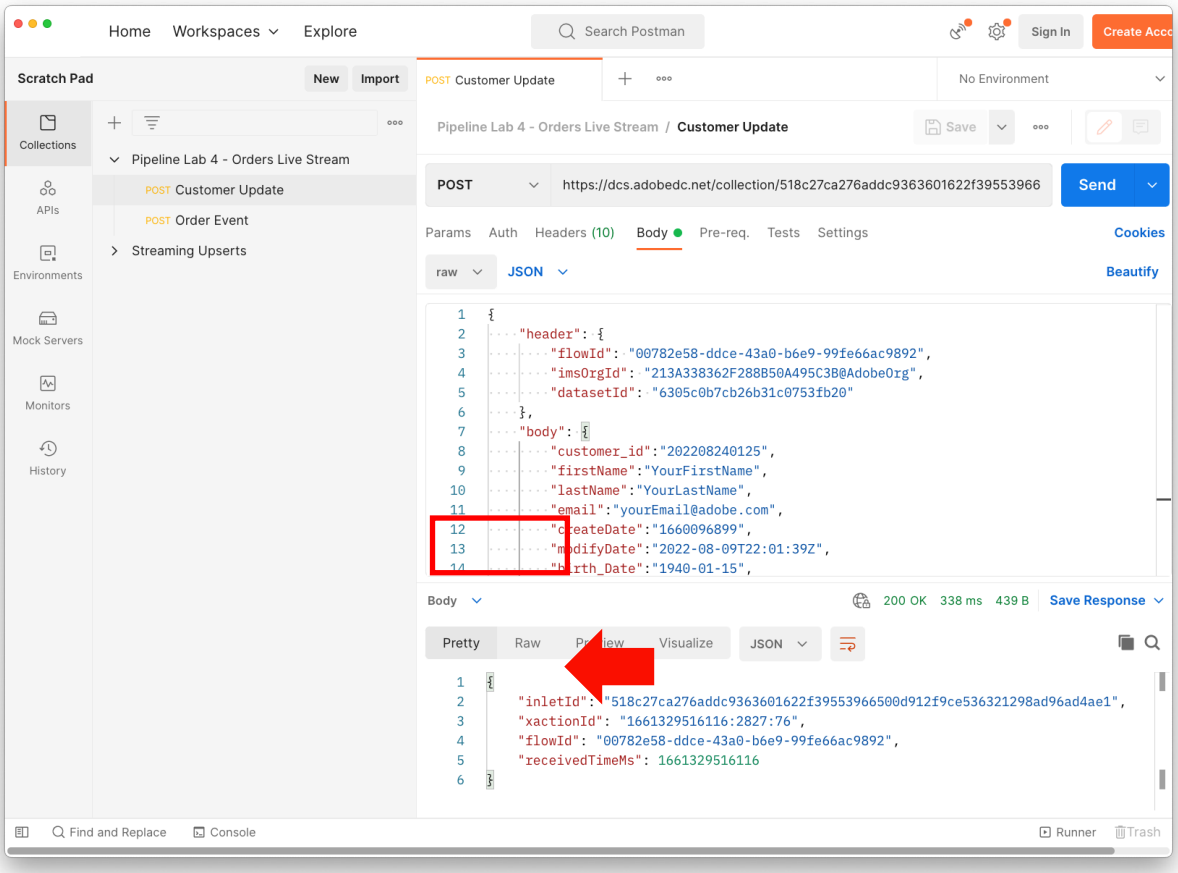
```
"body": {
    "customer_id":"202208240125",
    "firstName":"YourFirstName",
    "lastName":"YourLastName",
    "email":"yourEmail@adobe.com",
    "createDate":"1660096899",
    "modifyDate":"2022-08-09T22:01:39Z",
    "birth_Date":"1940-01-15",
    "mobile_phone":"309-115-7572",
    "email_optIn":"y",
    "sms_optIn":"n",
    "shipping_street_address":"71424 Messerschmidt Circle",
    "shipping_city":"Carol Stream",
    "shipping_state":"IL",
    "shipping_zip_code":"60351",
    "billing_street_address":"71424 Messerschmidt Circle",
    "billing_city":"Carol Stream",
    "billing_state":"IL",
    "billing_zip_code":"60351",
    "plan_id":"m1",
    "plan_name":"basic",
    "account_create_date":"Created on 2022-04-20T22:19:03Z",
    "account_end_date":"2022-01-20T13:15:32Z",
    "source":"inStore"
}
```

Change the following attributes in the JSON to make it unique:

Attribute Name	Type	Suggested values
customer_id	Number	Use your attendee number with date/timestamp to make it unique
firstName	String	Your First name
lastName	String	Your last name / sur name / initial
Email	Email	Must be a valid email in with proper email format.

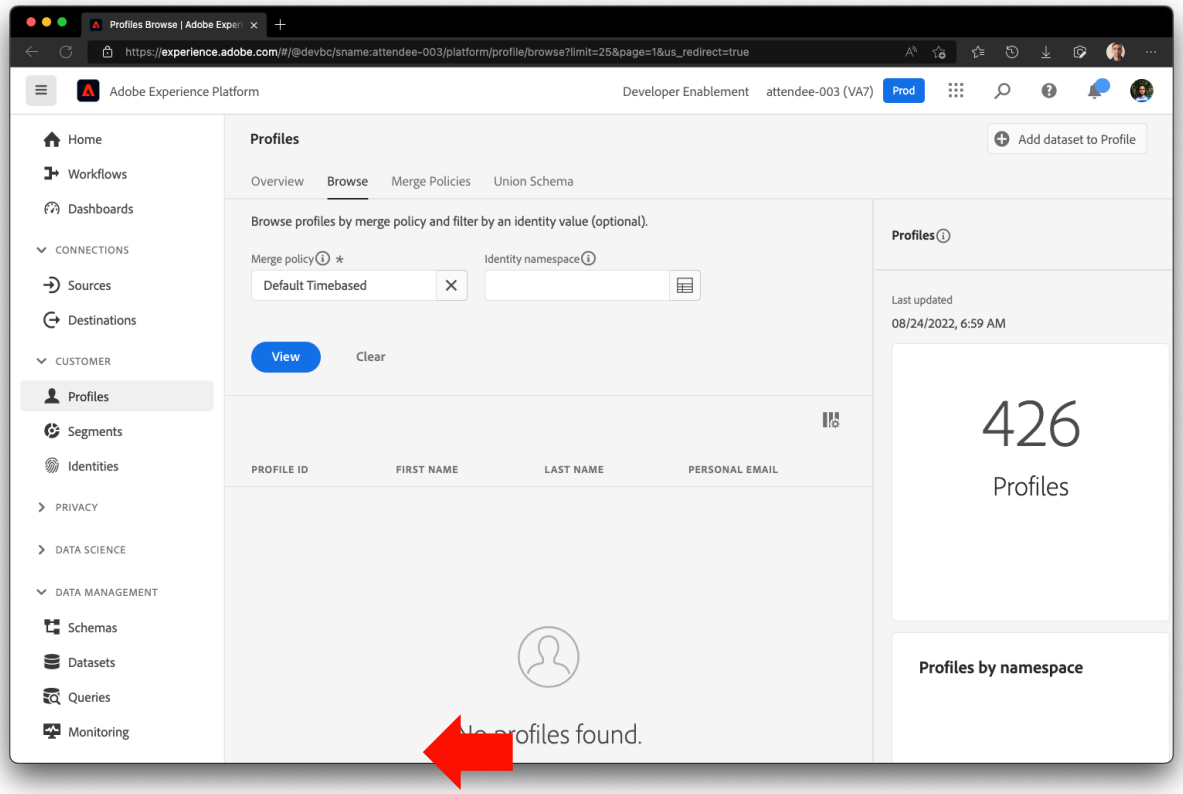
WARNING	Make sure these attributes are unique. If the row is not unique, you may be overriding someone else's data or vice versa.

Click Send to ingest the data. You should receive a **200 OK** from the Platform.

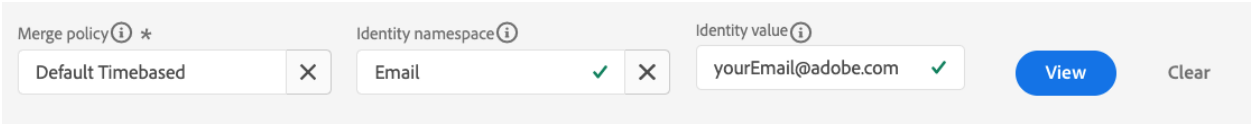


3.6. Verify the data in Profile

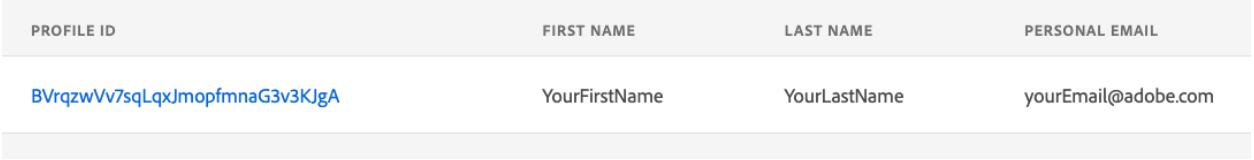
Go to **Profiles** in the Left Navbar and click on **Browse** tab.



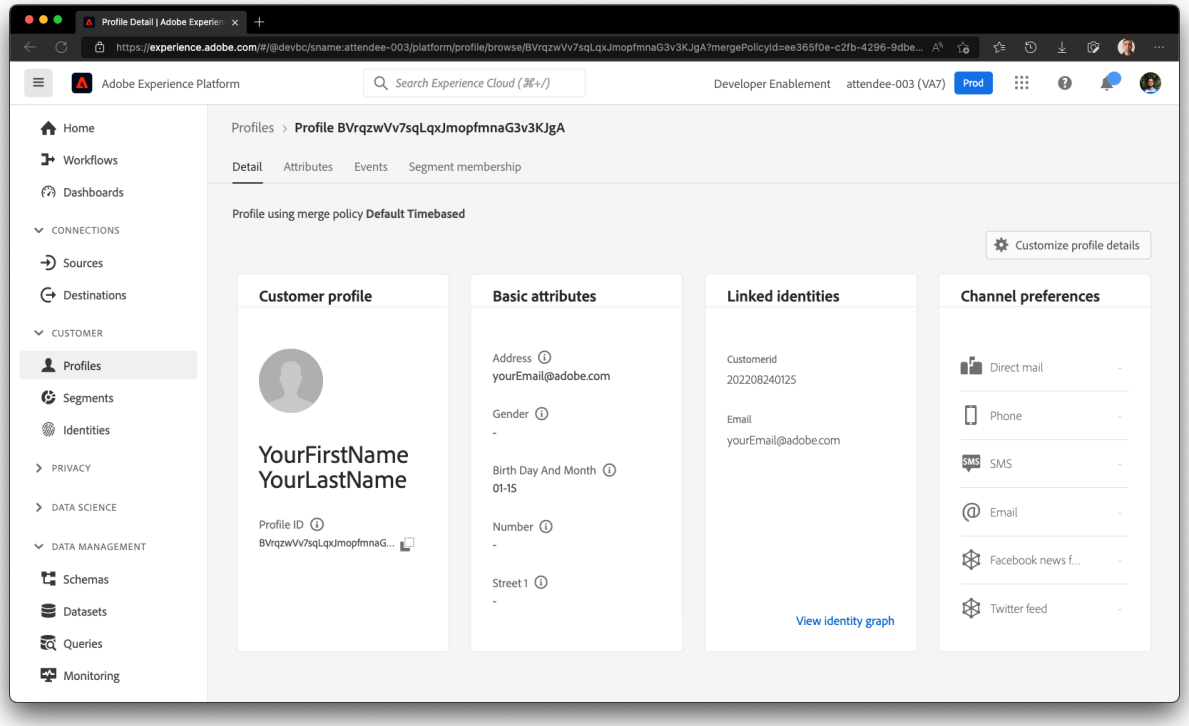
In the Browse tab, Leave the Merge policy to be the **Default Timebased** and select identity namespace as **Email**. Identity value text box appears. Type in your email (the one you used in the REST API Payload such as **yourEmail@adobe.com**) and press **View**.



A profile row will appear below. The profile row will have the first name, last name and email you supplied in the REST API call. Click on the Profile ID to open the Profile.



Notice the **person.birthDayAndMonth** populated in the Profile.



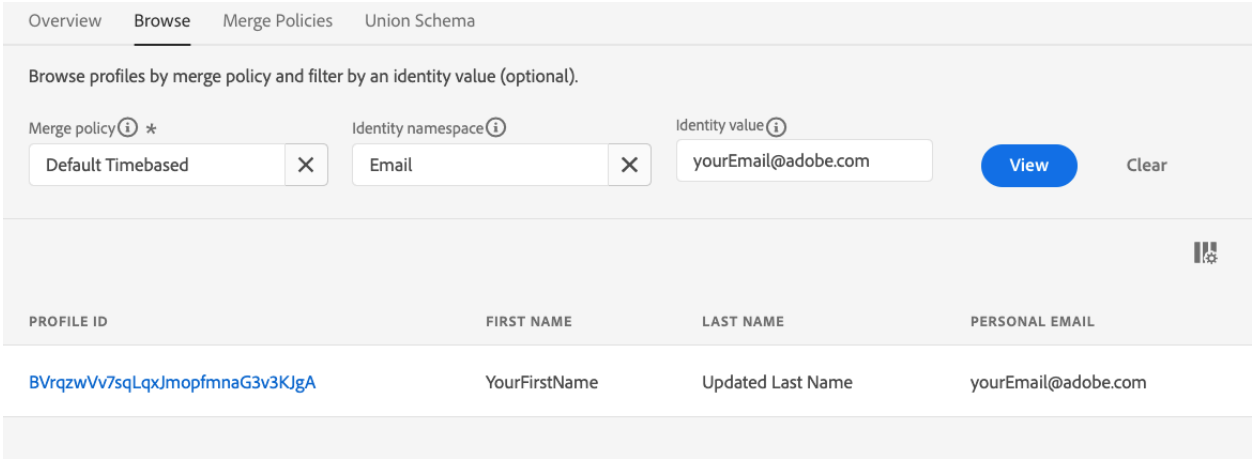
3.7.Learnings

1. We can always turn a batch ingestion flow in to Streaming flow by creating a new streaming connection and reusing the Mapping step from the batch flow
2. The fix we applied for **person.birthDayAndMonth** in Lab 2 is successful
3. When an update or new customer row arrived in the Streaming ingestion flow, it was immediately transformed into XDM and passed on to Profile. This demonstrates the Real-time nature of profile.

3.8. Additional exercises

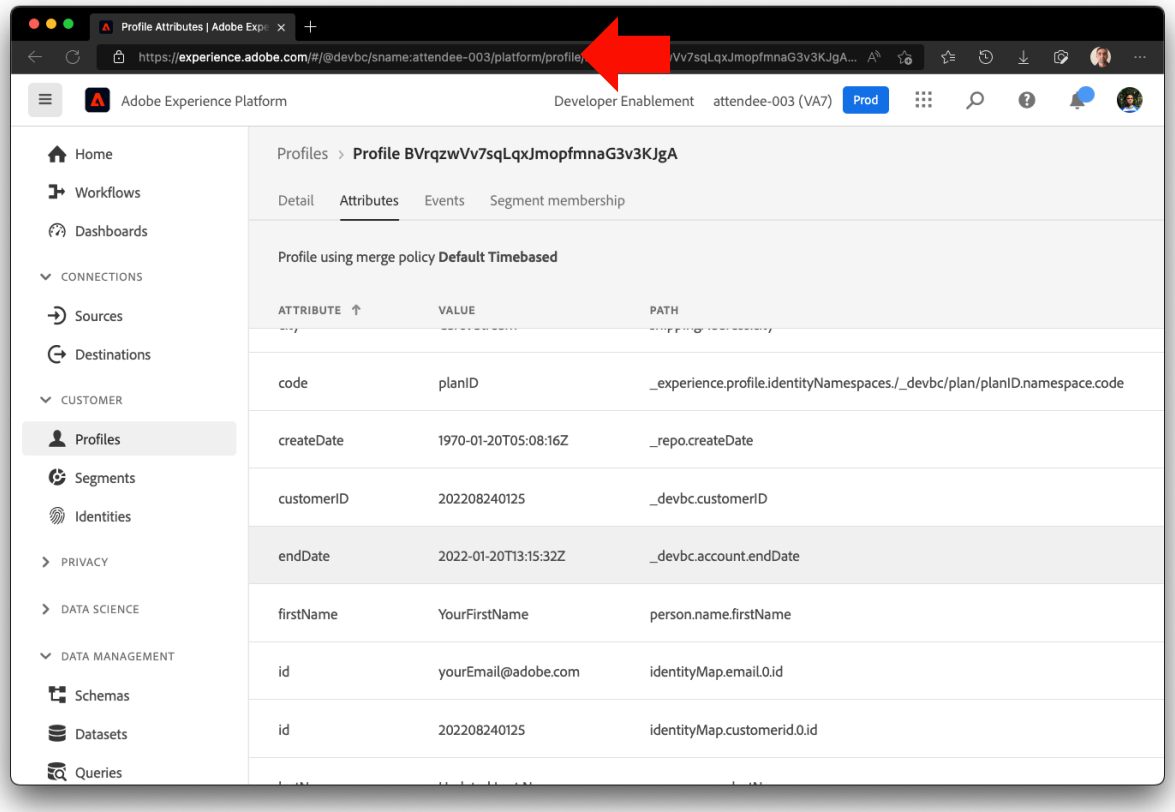
Exercise #1

Try executing the same REST API call with a slight name to the first name / last name. You will notice that the changes reflect in Profile in matter of minutes. Do NOT change the **Customer ID** or **email** attributes as they are identities and profile relies on them. It may take up to 15 minutes for the Profile values to be updated



Exercise #2

In the Attributes tab of the Profile, you will notice that only account end date is populated. But account create date is NOT populated. This is because we never fixed the issue with account create date in Data Prep Mapper step. Remember, that the account create date contained values such as "**Created on 2022-04-20T22:19:03Z**" whereas accepted values are similar to "**2022-04-20T22:19:03Z**"



Wait here for the instructor before proceeding further

4. Lab Tasks B – Orders – Live data (Streaming)

In this exercise, we will load the Orders data from Streaming source to AEP Data Lake and Profile.

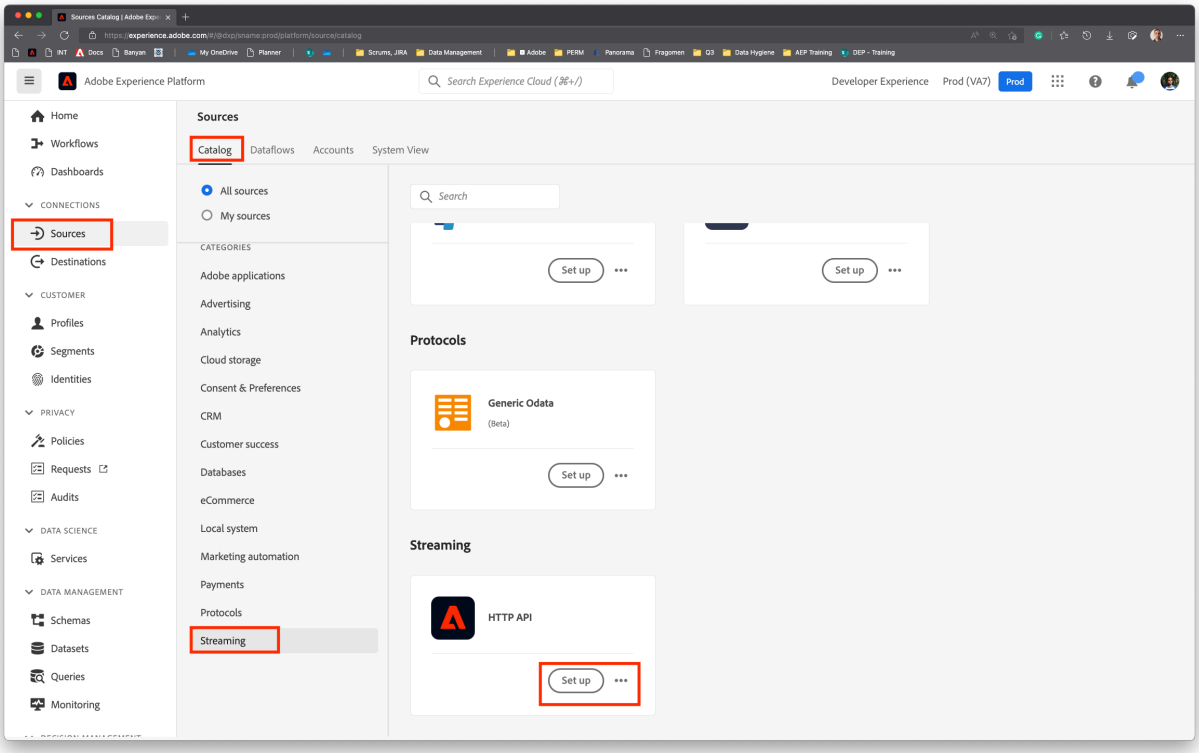
Pre-requisites

1. Orders JSON file is downloaded from Azure ADLS and available on your local system
2. Orders Schema and Dataset are already created

Steps

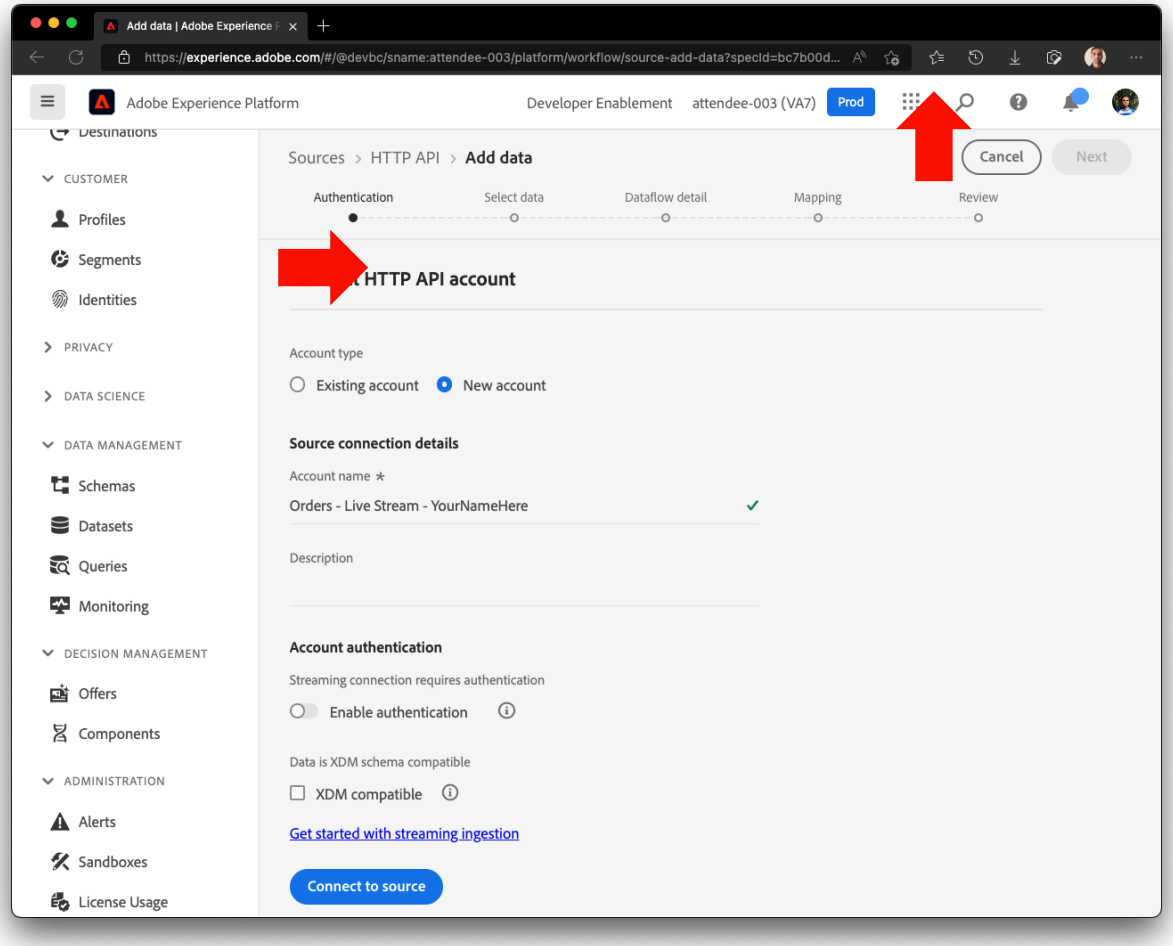
Go to Adobe Experience Platform → **Sources** → **Catalog** → **Streaming**. Click on **Setup** / **Add Data** for the HTTP API.

Tip	If at least one connection exists for that source, you will see "**Add data**" as the default action. If no connections exist for that source, you will see "**Setup**" as the default action.

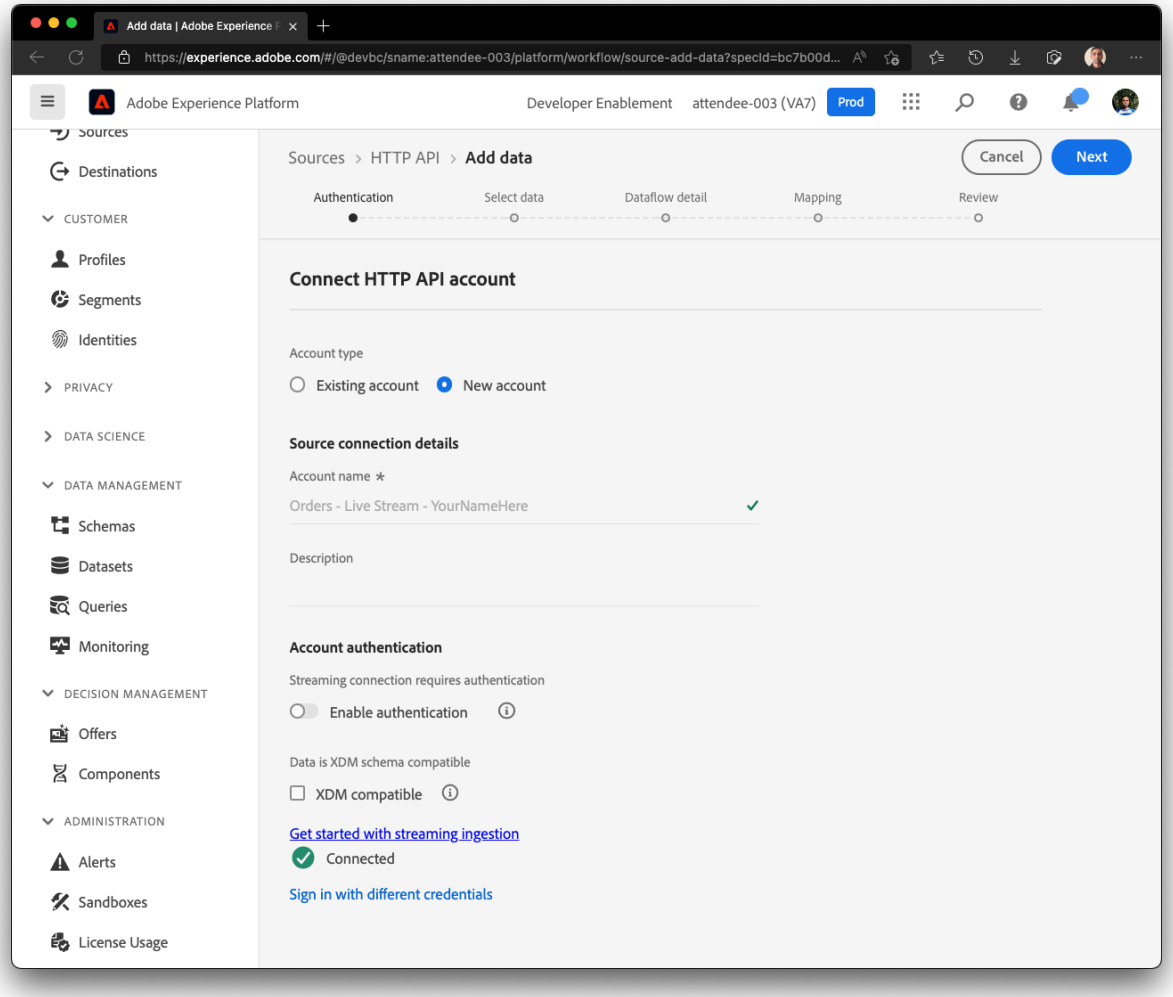


4.1. Create a Streaming Inlet

In the Streaming Source, create a **New account** as shown in the screenshot below. For the new account, type the name as **Orders Live Stream**. Click **Connect to source**. Enable authentication must be OFF (default). XDM Compatible must be unchecked (default).

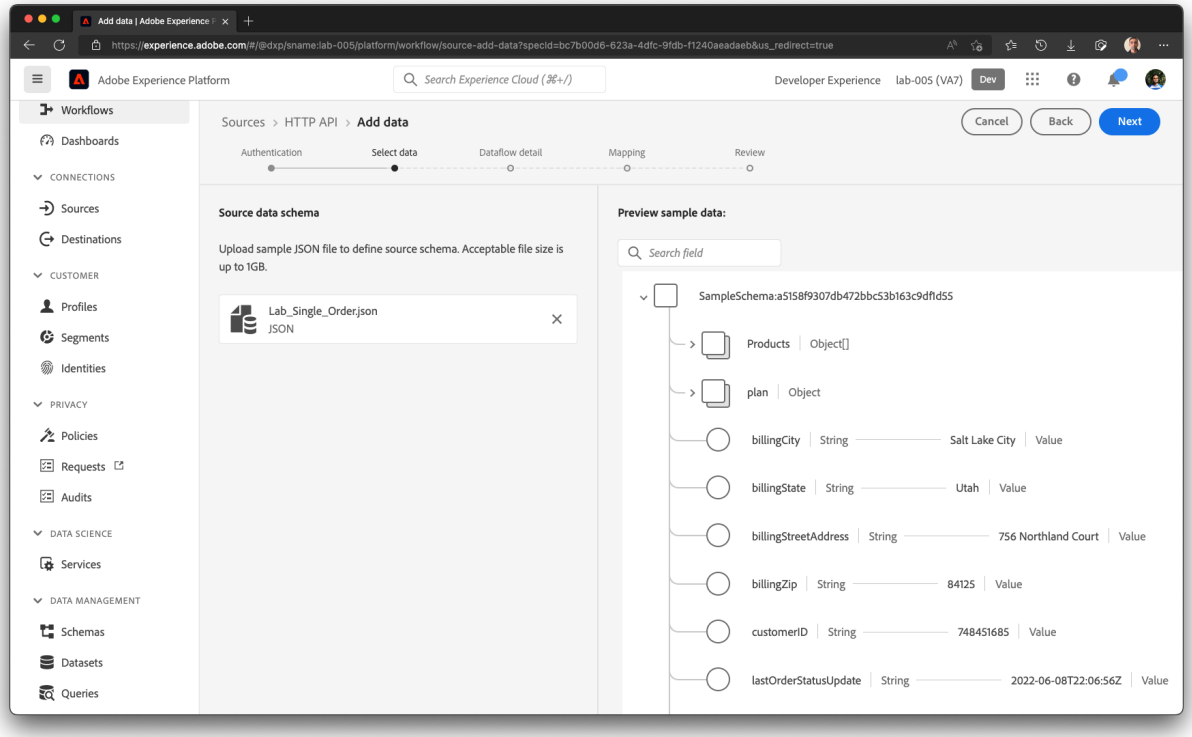


Once account is created, **Connected** message is displayed at the bottom. Click **Next** on the top right to continue.



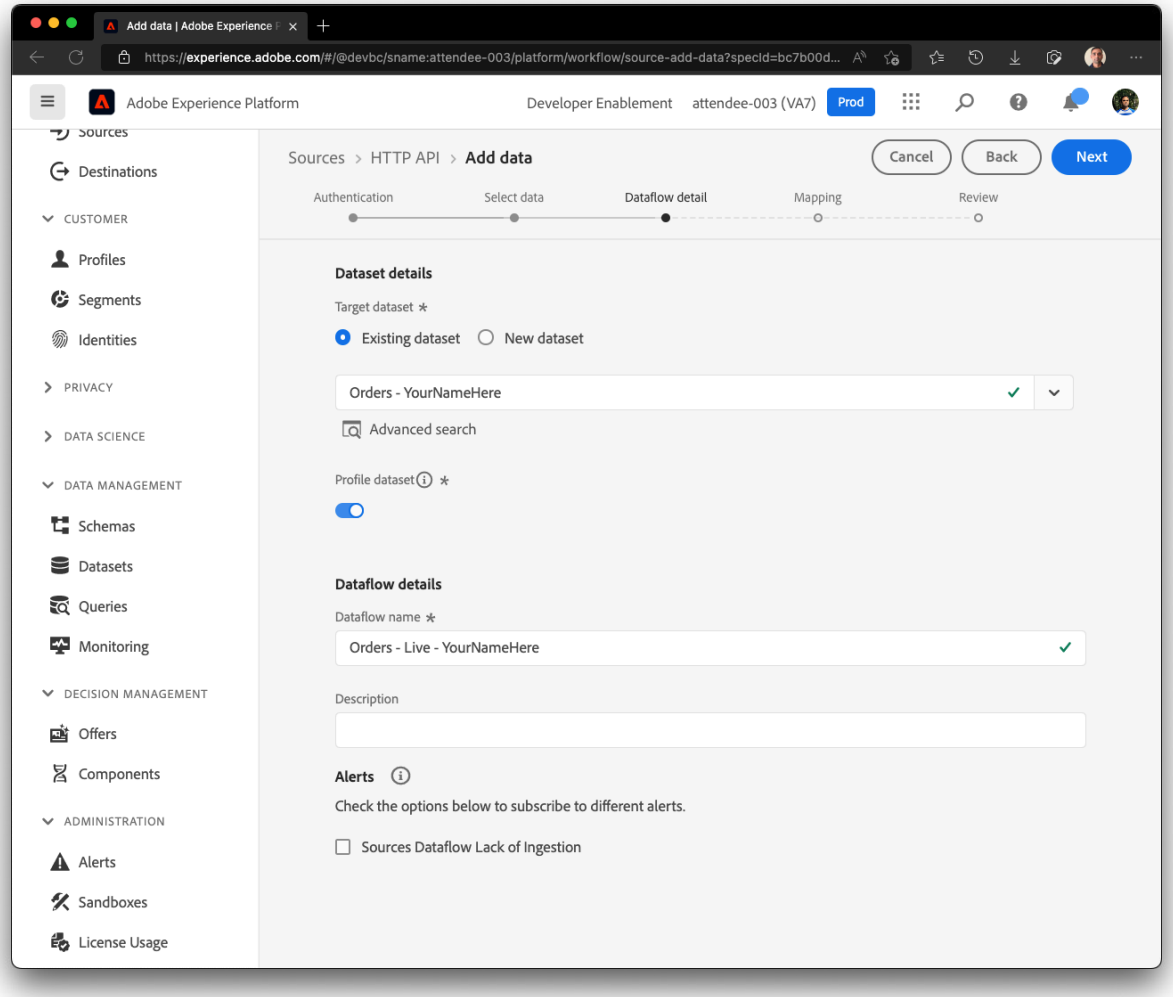
4.2. Upload sample data

Click **Upload files** and select the Orders JSON file (**Lab_Single_Order.json**) from local file system. Data Preview loads automatically. Click **Next**.



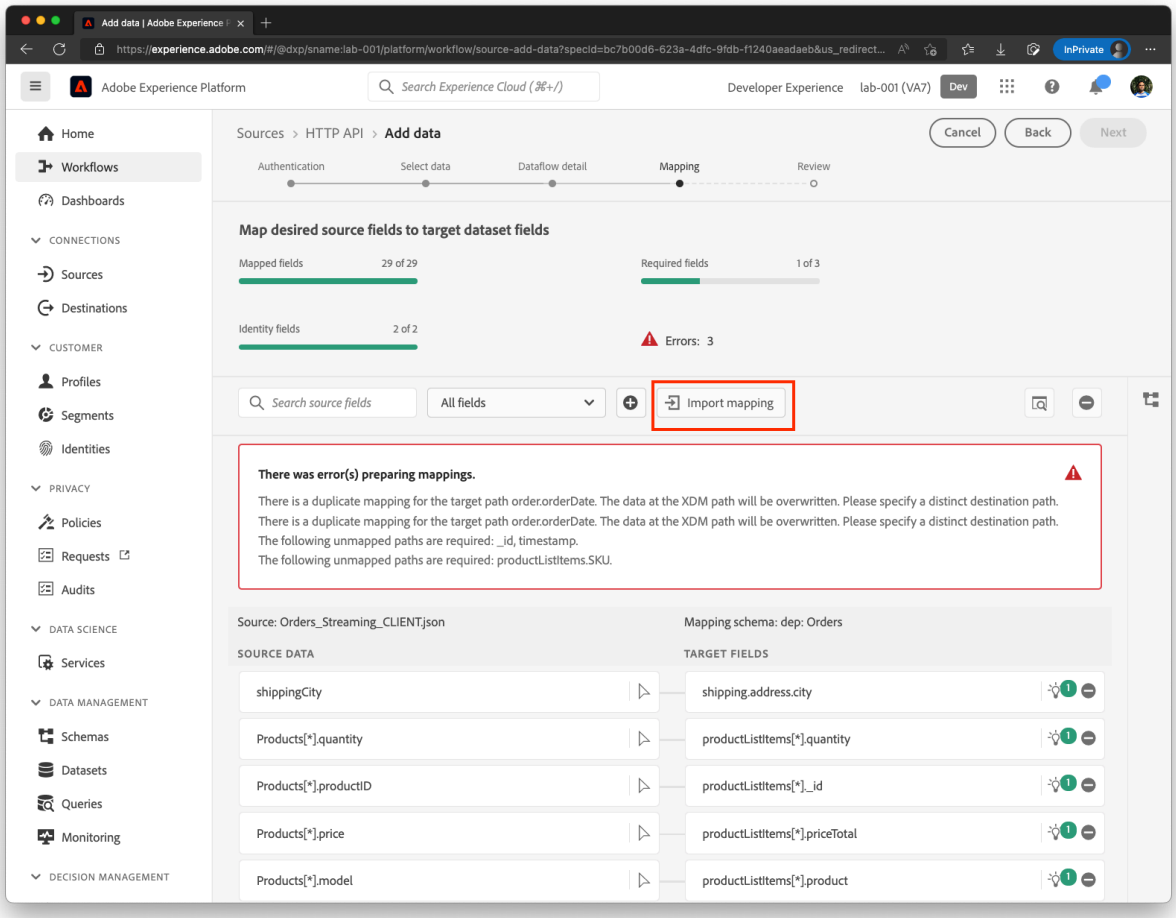
4.3. Define the Target dataset

In the screen, select existing dataset and choose **Orders learner - YourNameHere**. Set the Dataflow name as **Orders – Live Stream - YourNameHere**. Enable the Dataset for Profile. By toggling the **Profile dataset** box.

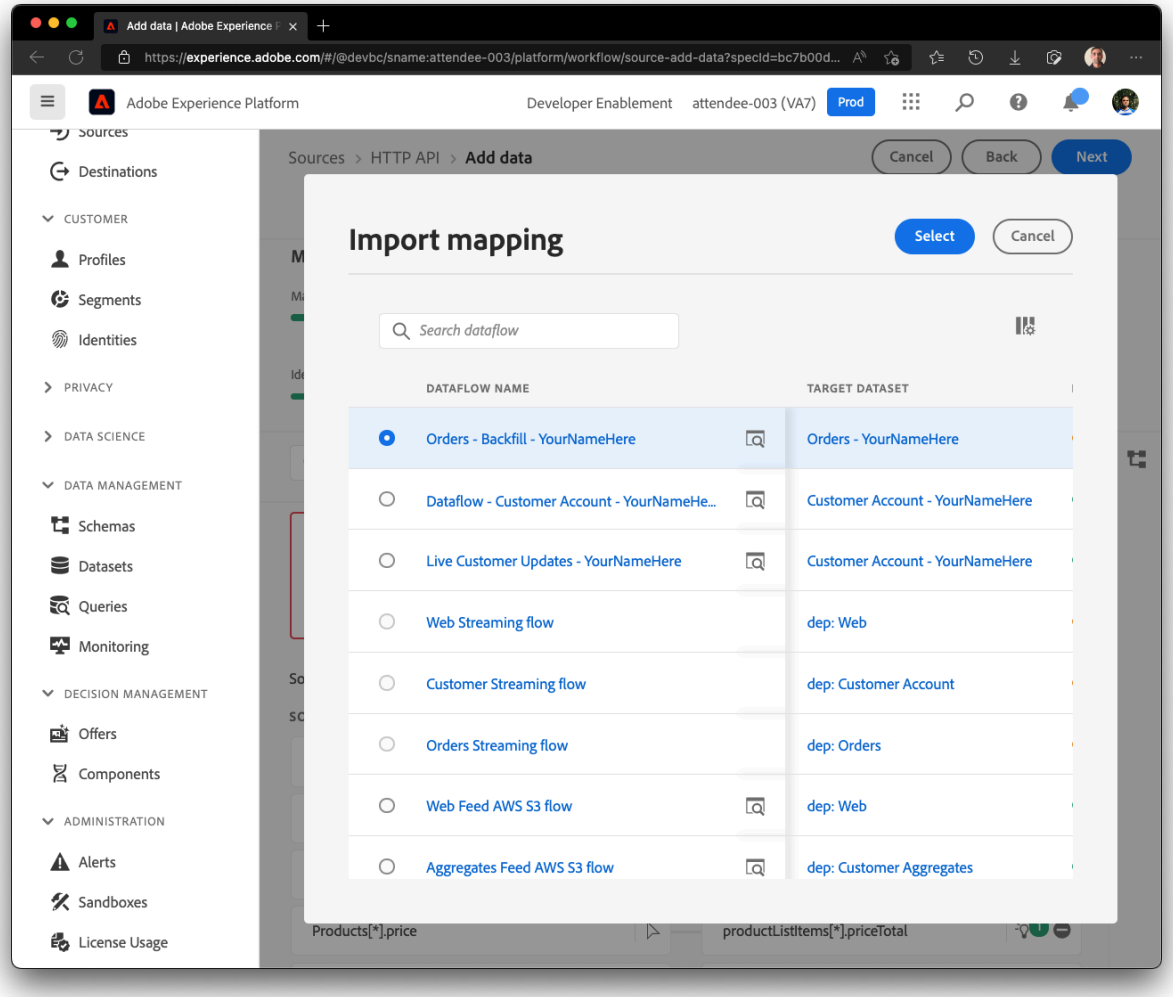


4.4. Data Prep / Transformation

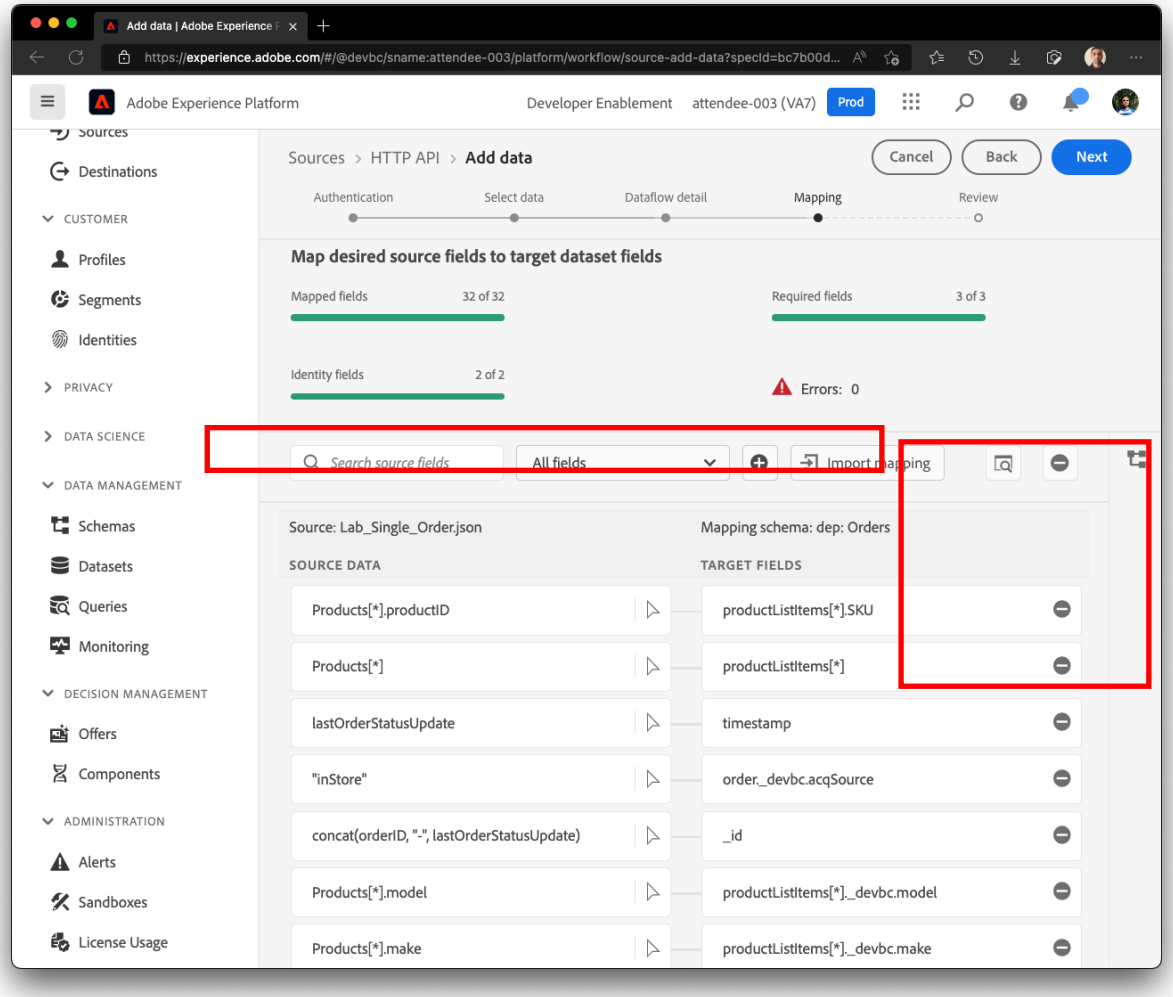
In the Data Prep (mapping) screen, Click on Import mapping button



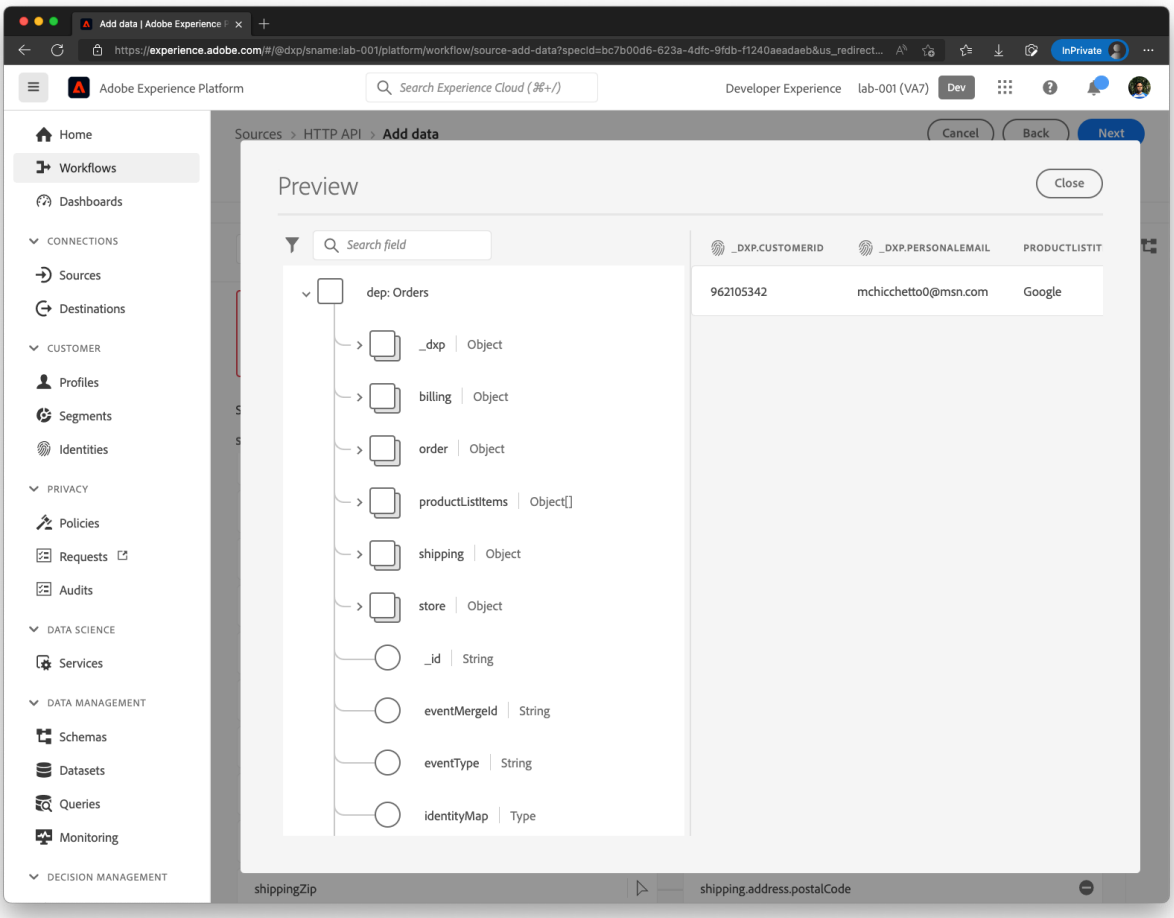
Select **Orders – Backfill - YourNameHere** and click **Select**.



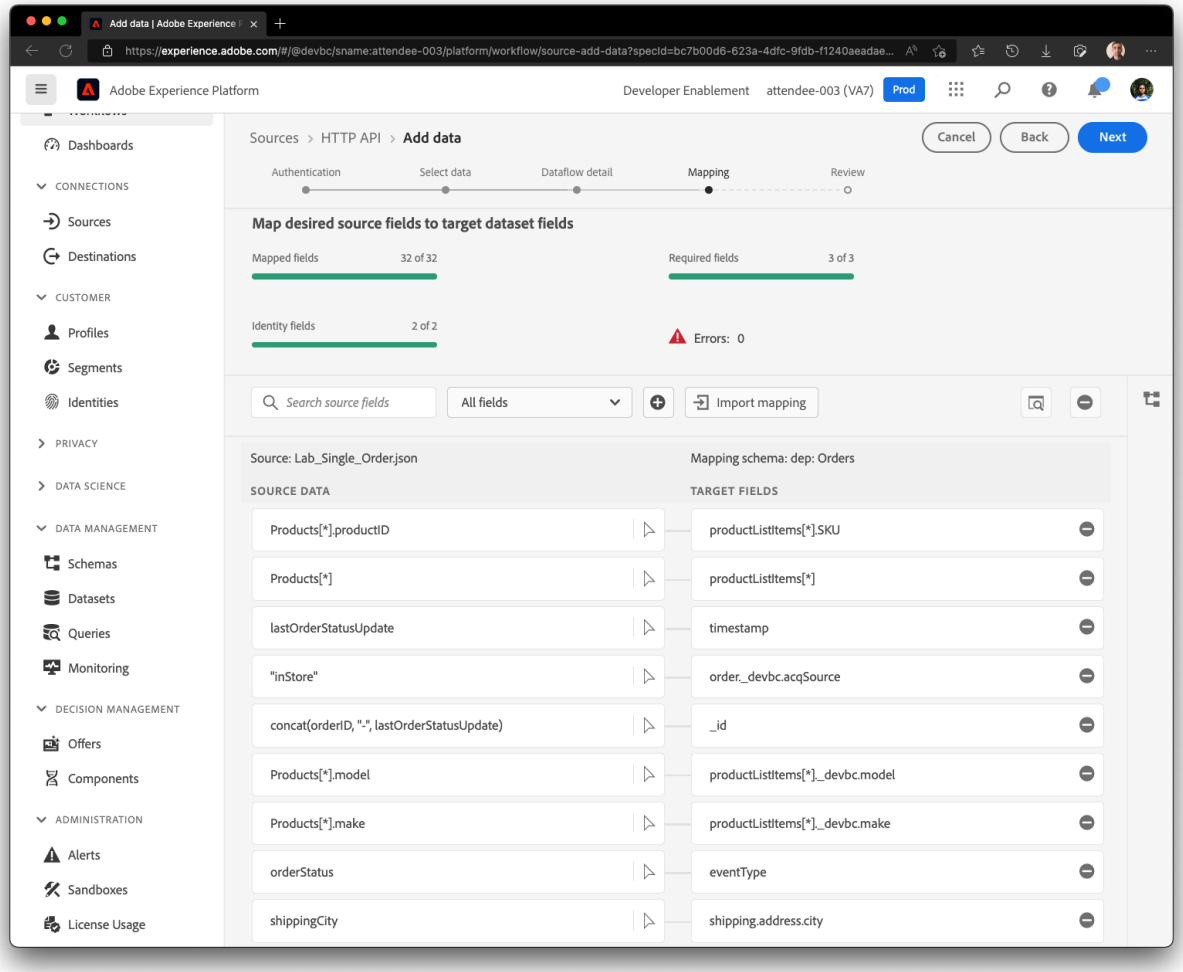
Mapping is now imported. All the previously existing mappings are Removed. Click Next.



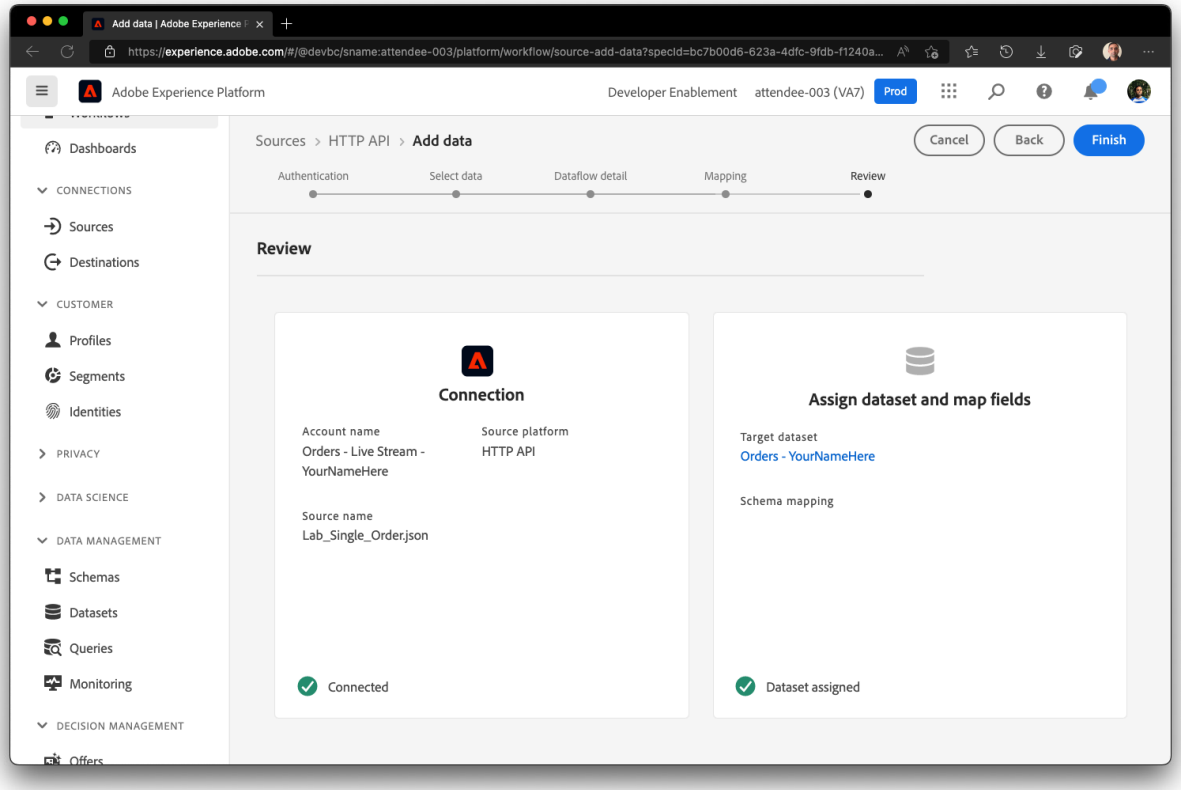
Click on the icon to Preview data button to view the mapping output. Ensure there are no red exclamations next to any attributes



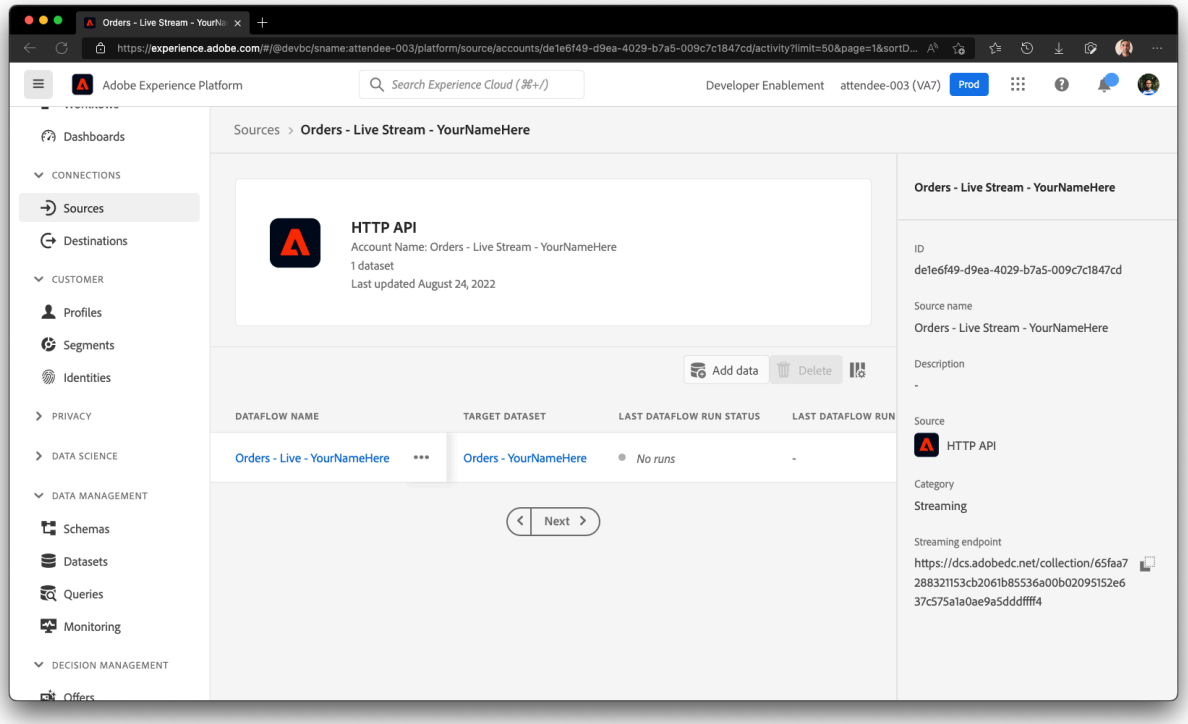
The final mapping set will look similar to this



Click Next to continue, review the Dataflow and save it by clicking Finish.



The Dataflow is now created. Ensure the status of the Dataflow is **Enabled** so that it is ready to receive events.



4.5. Ingest Live Data

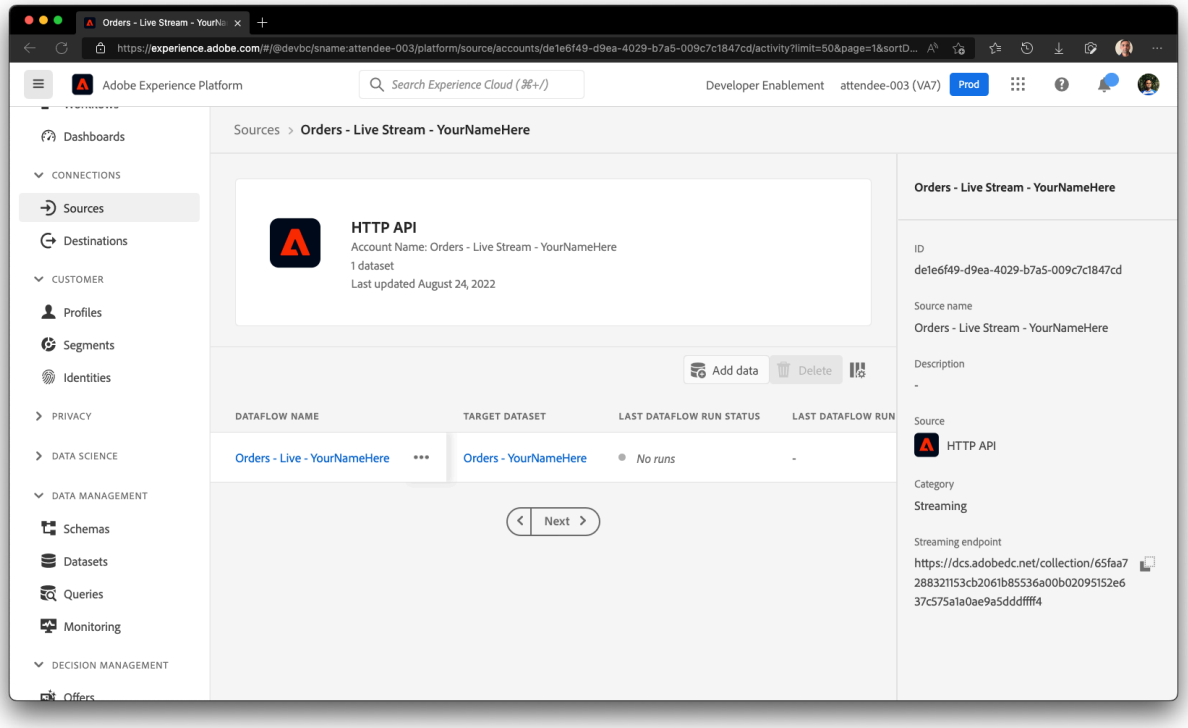
We will ingest the live data using a REST API call to our HTTP API Endpoint. To do this, we will need the following:

1. Streaming Endpoint
2. Dataflow ID
3. IMS Org ID
4. Dataset ID

Get the Streaming Endpoint

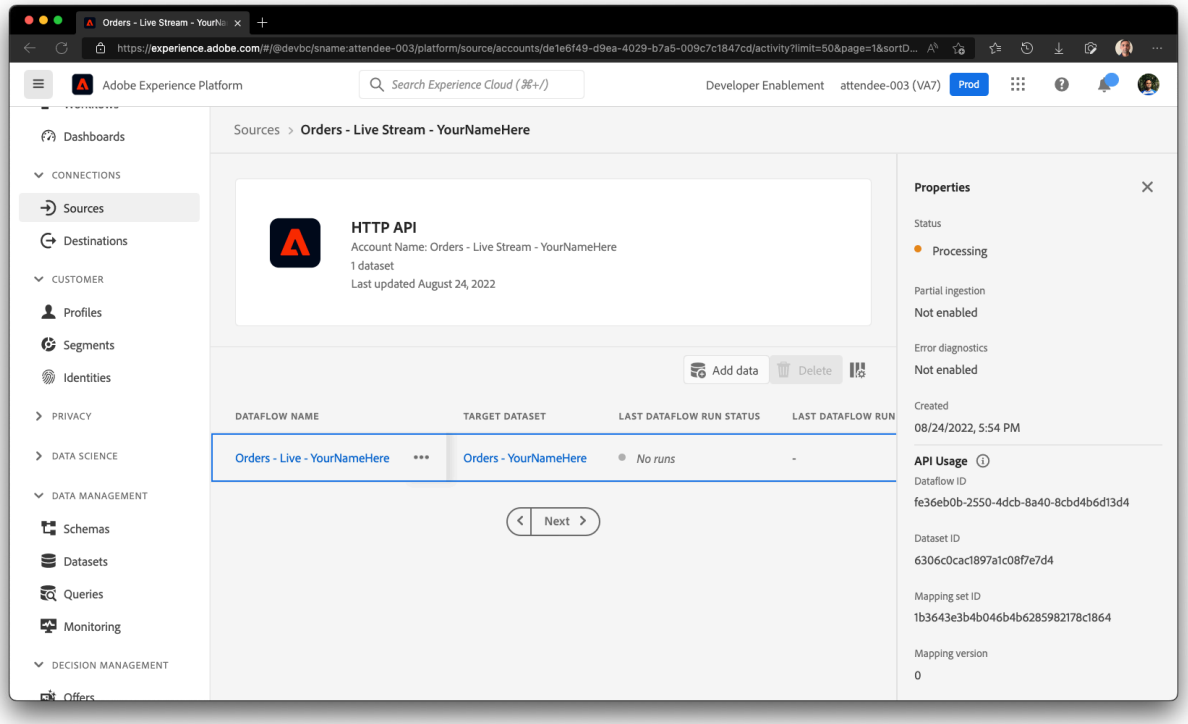
In the current screen, right hand side, note the Streaming endpoint.

Tip	If you do not see this information, most likely the Dataflow is selected on the left. Click on the Dataflow row (not the blue link) on the left again to unselect it.



Get Dataflow ID and Dataset ID

In the current screen, select the dataflow name, scroll down the right hand panel and take a note of the Dataflow ID and Dataset ID from the **API Usage** section.



4.6. Place a REST API call

In the Postman UI, set the following

Type: **POST**

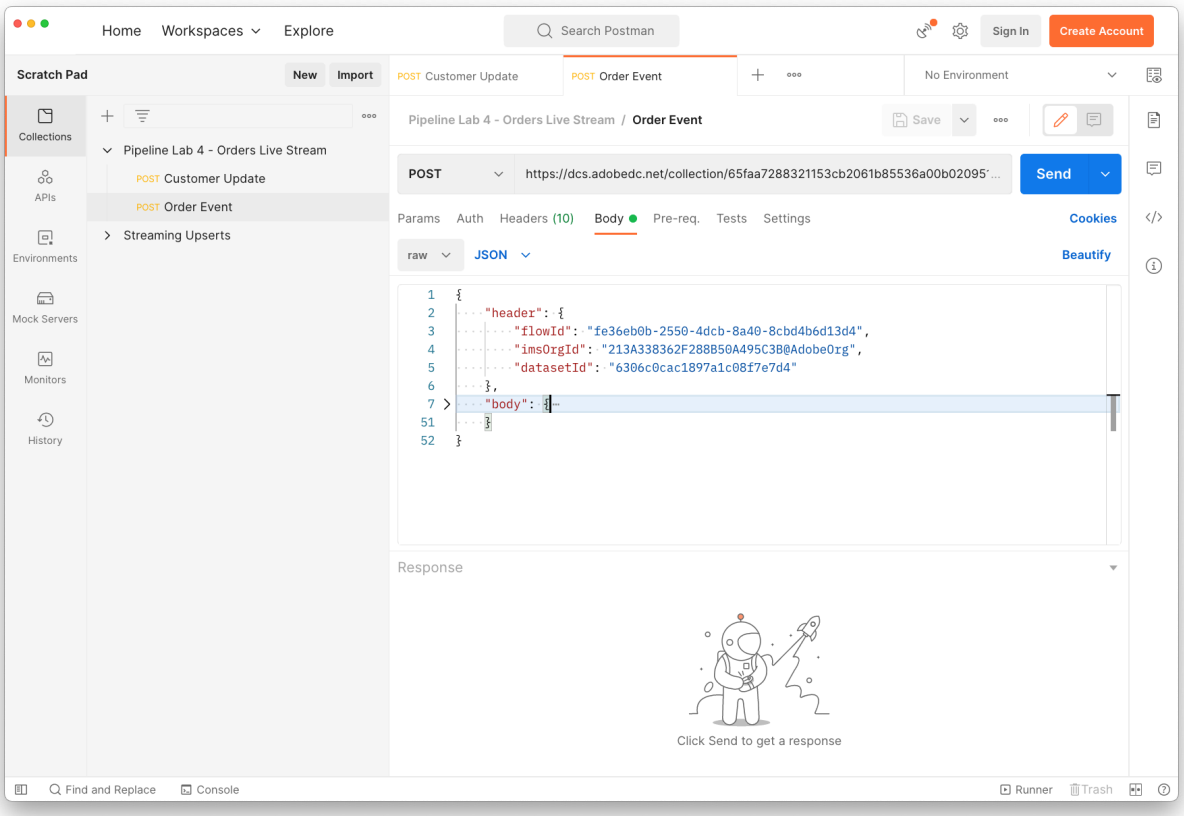
URL: « Streaming Endpoint » you captured in the previous step.

Body: The below structure.

Replace your flowId, and datasetId.

```
{
  "header": {
     "flowId": "fe36eb0b-2550-4dcb-8a40-8cbd4b6d13d4",
     "imsOrgId": "213A338362F288B50A495C3B@AdobeOrg",
     "datasetId": "6306c0cac1897a1c08f7e7d4"
  },
 "body": {

 }
}
```



Add the JSON request body as follows.

```
"body": {
        "orderStatus": "placed",
        "lastOrderStatusUpdate": "2022-08-11T17:43:01Z",
        "customerID": "202208240125",
        "personalEmail": "yourEmail@adobe.com",
        "orderID": "unique-yourEmail-order-001",
        "orderDate": "2022-08-11T17:43:01Z",
        "orderTotal": 291.02,
        "paymentType": "credit",
        "paymentAmount": 349.35,
        "paymentCurrencyCode": "USD",
        "paymentTransactionID": "24596662-1a9b-4fc1-a98f-b5ca687c523f",
        "storeID": "STORE-35",
        "shippingStreetAddress": "30133 Golf Parkway",
        "shippingCity": "Fullerton",
        "shippingState": "California",
        "shippingZip": "92640",
        "shippingMethod": "Standard",
        "shippingAmount": 0,
        "shippingDestination": "store",
        "billingStreetAddress": "497 Kensington Drive",
        "billingCity": "Glinojeck",
        "billingZip": "06-450",
        "plan": {
          "ID": "m3"
        },
        "Products": [
          {
            "productID": "PRODUCT-6",
            "quantity": 1,
            "price": 150.23,
            "currencyCode": "USD",
            "model": "v2",
            "make": "Google"
          },
```

```
      {
        "productID": "PRODUCT-11",
        "quantity": 1,
        "price": 150.23,
        "currencyCode": "USD",
        "model": "v1",
        "make": "Google"
      }
    ]
}
```

Change the following attributes to make it unique:

Attribute Name	Type	Suggested values
customerID	Number	1. Must be valid number 2. MUST Match what you used in Customer Account REST API call
personalEmail	Email	1. Must be valid email with proper format. 2. MUST match what you used in Customer Account REST API call
orderID	String	1. MUST be unique for every REST API call you make. 2. Suggested to have your email suffixed by a sequence number
last Order StatusUpdate	Timestamp	1. Must be a valid timestamp 2. Example format is 2022-08-11T17:43:01Z 3. MUST be current or past date

WARNING	Make sure these attributes are unique. If the row is not unique, system will not warn and the event will be ignored (not reported as warning / error).

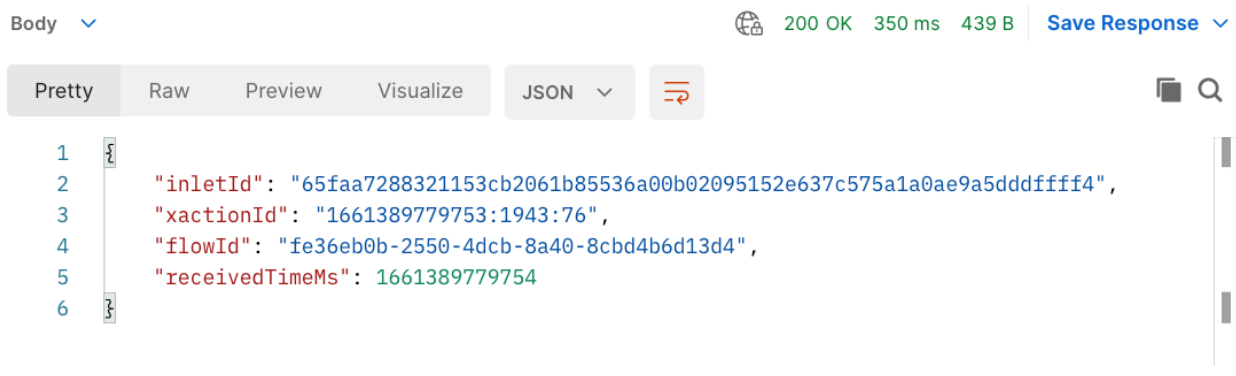
Full REST call will look like this:

```
{
  "header": {
    "flowId": "fe36eb0b-2550-4dcb-8a40-8cbd4b6d13d4",
    "imsOrgId": "213A338362F288B50A495C3B@AdobeOrg",
    "datasetId": "6306c0cac1897a1c08f7e7d4"
  },
  "body": {
    "orderStatus": "placed",
    "lastOrderStatusUpdate": "2022-08-11T17:43:01Z",
    "customerID": "202208240125",
    "personalEmail": "yourEmail@adobe.com",
    "orderID": "unique-yourEmail-order-001",
    "orderDate": "2022-08-241T17:43:01Z",
    "orderTotal": 291.02,
    "paymentType": "credit",
    "paymentAmount": 349.35,
    "paymentCurrencyCode": "USD",
    "paymentTransactionID": "24596662-1a9b-4fc1-a98f-b5ca687c523f",
    "storeID": "STORE-35",
    "shippingStreetAddress": "30133 Golf Parkway",
    "shippingCity": "Fullerton",
    "shippingState": "California",
    "shippingZip": "92640",
    "shippingMethod": "Standard",
    "shippingAmount": 0,
    "shippingDestination": "store",
    "billingStreetAddress": "497 Kensington Drive",
    "billingCity": "Glinojeck",
    "billingZip": "06-450",
    "plan": {
      "ID": "m3"
    },
    "Products": [
      {
        "productID": "PRODUCT-6",
```

```
            "quantity": 1,
            "price": 150.23,
            "currencyCode": "USD",
            "model": "v2",
            "make": "Google"
        },
        {
            "productID": "PRODUCT-11",
            "quantity": 1,
            "price": 150.23,
            "currencyCode": "USD",
            "model": "v1",
            "make": "Google"
        }
    ]
  }
}
```

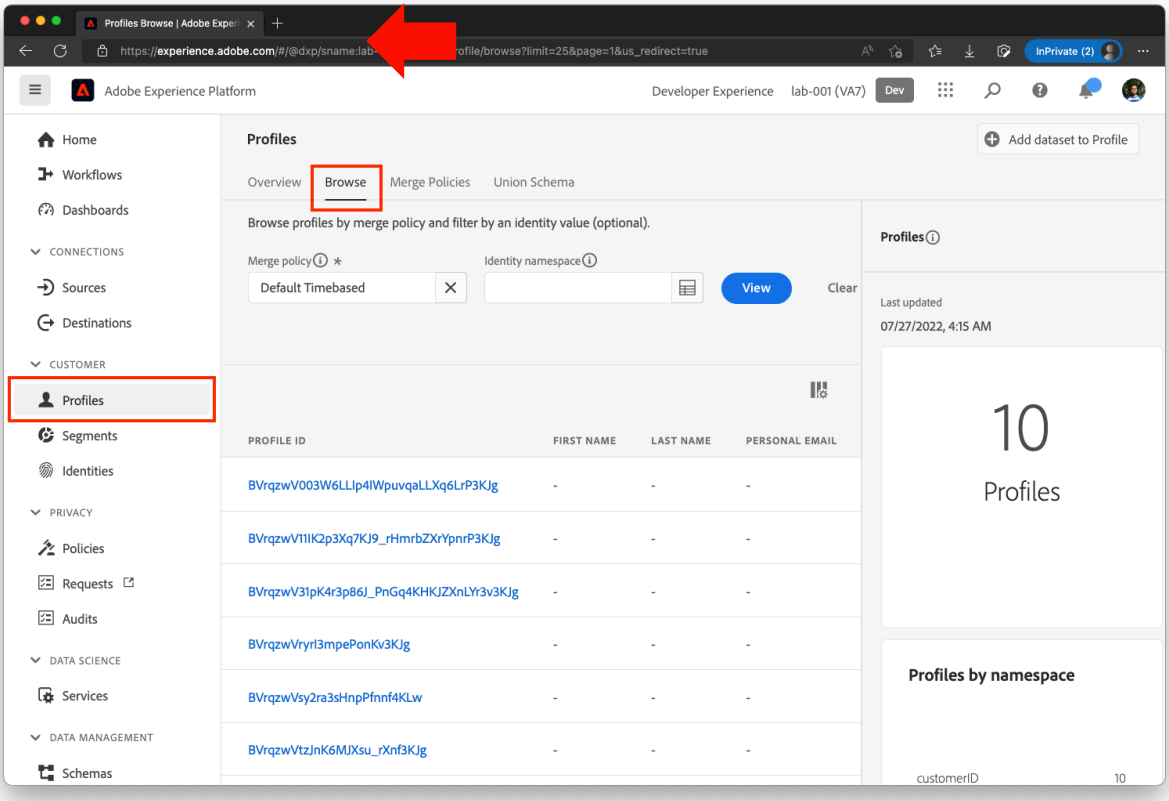
Click Send and the event will be posted to the Streaming Endpoint. The response body will be similar to this:

```
{
  "inletId": "65faa7288321153cb2061b85536a00b02095152e637c575a1a0ae9a5ddddffff4",
  "xactionId": "1661389779753:1943:76",
  "flowId": "fe36eb0b-2550-4dcb-8a40-8cbd4b6d13d4",
  "receivedTimeMs": 1661389779754
}
```

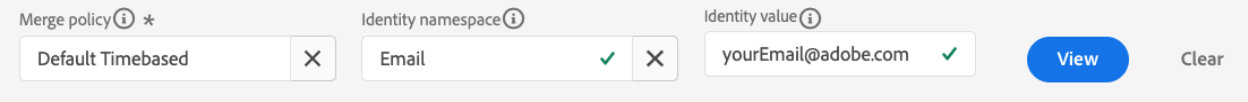


4.7. Verify data in Profile

Go to Adobe Experience Platform → **Profiles** in the left Nav bar and click on the Browse on the top.



In the Browse tab, Leave the Merge policy to be the **Default Timebased** and select identity namespace as **Email**. Identity value text box appears. Type in your email (the one you used in the REST API Payload such as **yourEmail@adobe.com**) and press **View**.



A profile row will appear below. The profile row will have the first name, last name and email you supplied in the REST API call. Click on the Profile ID to open the Profile.

PROFILE ID	FIRST NAME	LAST NAME	PERSONAL EMAIL
BVrqzwVv7sqLqxJmopfmnaG3v3KJgA	YourFirstName	Updated Last Name	yourEmail@adobe.com

Profile gets loaded. This profile was originally loaded via Customer Accounts Lab (Lab 2) and updated in the previous Customer Accounts REST call in this lab. Click on the Events tab.