



# Student Workbook

## Getting Started with Adobe Experience Platform

Adobe Copyright Protected

©2022 Adobe. All rights reserved.

## Getting Started with Adobe Experience Platform

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe. Adobe assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names in sample templates are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe, the Adobe logo, Acrobat, the Creative Cloud logo, and the Adobe Marketing Cloud logo are either registered trademarks or trademarks of Adobe in the United States and/or other countries.

All other trademarks are the property of their respective owners.

Adobe, 345 Park Avenue, San Jose, California 95110, USA.

Notice to U.S. Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

05122022

# Contents

|   |           |
|---|-----------|
| <b>Module 1: Introduction to Adobe Experience Platform</b>                              | <b>5</b>  |
| Introduction  | 5         |
| Objectives  | 5         |
| Experience Platform: Features and Benefits  | 6         |
| Experience Platform: Architecture   | 8         |
| Adobe Experience Platform and Adobe Cloud Applications Integration                      | 10        |
| Application Services Powered by Experience Platform: An Overview                        | 12        |
| Exercise 1: Navigate through the Experience Platform UI                                 | 13        |
| <b>Module 2: Managing Data in Adobe Experience Platform</b>                             | <b>28</b> |
| Introduction  | 28        |
| Objectives  | 28        |
| Data Ingestion Pipeline in Experience Platform  | 29        |
| Data Ingestion Methods in Experience Platform   | 31        |
| XDM: An Overview  | 32        |
| Data Governance and Privacy Service in Experience Platform                              | 34        |
| Data Governance   | 34        |
| Privacy Service   | 35        |
| Exercise 1: Verify data ingestion from a website into Experience Platform               | 36        |
| Exercise 2: Ingest data by using a CSV file mapped to the XDM schema                    | 41        |
| Exercise 3: Apply data governance to a dataset  | 46        |
| <b>Module 3: Managing Real-Time Customer Profiles in Adobe Experience Platform</b>      | <b>49</b> |
| Introduction  | 49        |
| Objectives  | 49        |
| Real-Time Customer Profile: An Overview   | 50        |
| Identity Service  | 50        |
| Segmentation Service  | 51        |
| Exercise 1: Create a customer profile in Experience Platform                            | 52        |
| (Optional) Exercise 2: Authenticate Postman to Adobe I/O to execute API calls           | 57        |
| Exercise 3: Verify profiles in Experience Platform                                      | 66        |
| Exercise 4: Add cross-device data to Experience Platform                                | 68        |
| Exercise 5: Create a segment in Experience Platform by using the Experience Platform UI | 71        |
| (Optional) Exercise 6: Create a segment in Experience Platform by using API             | 75        |

|  |            |
|--|------------|
| <b>Module 4: Querying in Adobe Experience Platform</b>   | <b>80</b>  |
| Introduction   | 80         |
| Objectives   | 80         |
| Query Service in Experience Platform   | 81         |
| Adobe-Defined Functions  | 81         |
| Exercise 1: Navigate through the Query service UI in Experience Platform and execute a query                       | 82         |
| (Optional) Exercise 2: Connect PSQL to Experience Platform   | 86         |
| Exercise 3: Create queries for the specific XDM values in Experience Platform                                      | 89         |
| (Optional) Exercise 4: Create queries for the specific XDM values in PSQL  | 98         |
| Exercise 5: Create basic queries to analyze data in Experience Platform  | 101        |
| (Optional) Exercise 6: Create basic queries for data analysis in PSQL  | 106        |
| Exercise 7: Create queries by using ADF to obtain specific contextual values in Experience Platform                | 110        |
| (Optional) Exercise 8: Create queries in PSQL by using ADF to obtain specific contextual values                    | 117        |
| Exercise 9: Create a query combining online, call center, and loyalty data available in the Experience Platform UI | 120        |
| (Optional) Exercise 10: Create a query combining online, call center, and loyalty data available in PSQL           | 122        |
| Exercise 11: Create a dataset from the query result in the Experience Platform UI                                  | 123        |
| Exercise 12: Create a visualization report in Power BI Desktop by using the Experience Platform dataset            | 126        |
| <b>Appendix: Data Science Workspace in Adobe Experience Platform</b>   | <b>132</b> |
| Introduction   | 132        |
| Objectives   | 132        |
| Data Science Workspace: An Overview  | 133        |
| Creating a Machine Learning Model: The Process   | 134        |
| Preparing the data   | 134        |
| Authoring a recipe   | 134        |
| Training and evaluating a model  | 134        |
| Operationalizing the model   | 135        |
| Recipe Builder Notebook in JupyterLab  | 136        |
| Exercise 1: Analyze datasets in Experience Platform  | 137        |
| Exercise 2: Transform data into an Experience Platform dataset by using the JupyterLab notebook                    | 140        |
| Exercise 3: Load and analyze the training data   | 149        |
| Exercise 4: Create a recipe by using the JupyterLab notebook   | 152        |
| Exercise 5: Train and test a machine learning model  | 157        |
| <b>Appendix: Intelligent Services – Customer AI</b>  | <b>162</b> |
| Introduction   | 162        |
| Objectives   | 163        |
| Exercise 1: View Schema and Dataset Configuration  | 164        |
| Exercise 2: Set up a New Customer AI Instance  | 166        |
| Exercise 3: Visualize propensity score and create segments   | 168        |
| Additional Resources   | 170        |

## Module 1

# Introduction to Adobe Experience Platform

---

### Introduction

Customers interact with brands by using multiple devices, such as mobile phones, tablets, computers, and televisions for various transactions. They expect personalized and relevant experiences delivered to them in real time. It is challenging for businesses to reach the right audience with the right content at the right time.

Adobe Experience Platform provides an open, intelligent, real-time platform that accelerates the time to value (TtV). With Experience Platform, businesses can centralize and standardize all data, create dynamic real-time customer profiles, enrich customer profiles by using Artificial Intelligence (AI) and Machine Learning (ML) capabilities, and ensure proper governance of data.

### Objectives

After completing this module, you will be able to:

- Describe the features and benefits of Experience Platform
- Explain the Experience Platform architecture
- Explain how Experience Platform adds value to Adobe Experience Cloud
- Explain the Application Services powered by Experience Platform
- Navigate through the Experience Platform user interface (UI)

# Experience Platform: Features and Benefits

---

Experience Platform enables you to:

- Centralize and standardize customer data
- Deliver real personalization by using real-time customer profiles
- Govern and control the customer data usage for marketing actions
- Obtain deeper insights by querying behavioral data across products
- Create powerful, usable audience segments in real time

The features of Experience Platform are:

- Data ingestion: Assembles customer data from multiple sources into Experience Platform.
- Experience Data Model (XDM): Standardizes the customer data and defines the schemas for customer experience management
- Real-time Customer Profile: Delivers personalized experiences to the customers by using unified, complete, and centrally accessible profile
- Data Lake: Stores vast amount of customer data in the central repository for customer profile-based Azure data lake storage
- Experience Platform Launch: Connects multiple technologies and converts data into action
- AI and ML: Recognizes familiar customers on unfamiliar devices to understand how customers interact with your brand
- Segmentation Service: Builds and maintains audience segments based on real-time customer profiles
- Query Service: Retrieves the data from customer datasets and executes fast, petabyte-scale Structured Query Language (SQL) queries to discover the customer behavior and generate impactful insights using a business intelligence (BI) tool
- Data Science Workspace: Uses ML models powered by Adobe Sensei to derive insights and predictions
- Application Program Interface (API): Provides access to the developer tools that you need to build, integrate, and extend the Adobe products

- Location Service: Identifies key audiences, engages with the audiences, and understands customer visits by adding the location dimension to your analysis
- Auditor: Scans and audits pages and gets recommendations to improve your Adobe implementations
- Data Governance: Ensures the proper use of data within Experience Platform and when data is shared between systems
- Privacy Services: Automates compliance with the data privacy regulations
- Security: Protects your stored data with a secured infrastructure of components and services

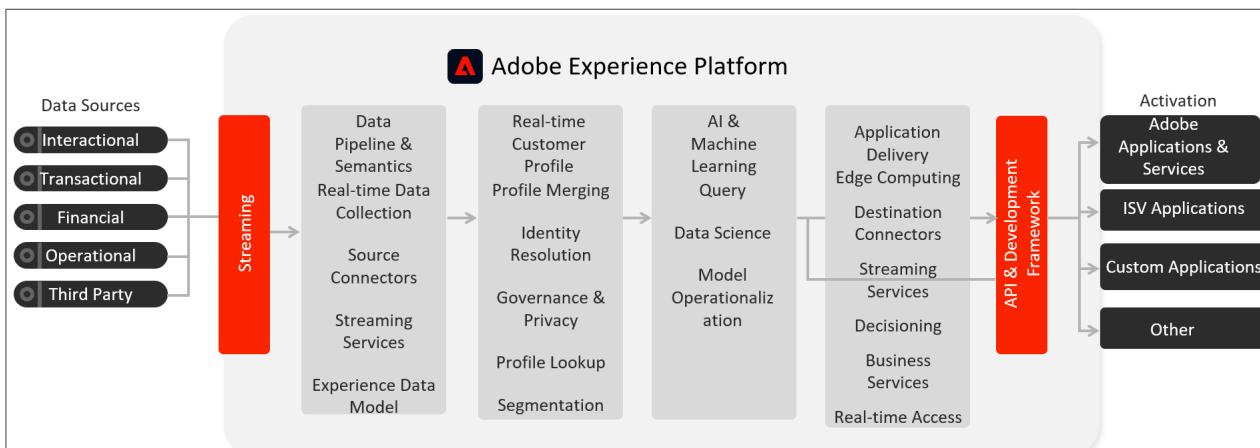
# Experience Platform: Architecture

Experience Platform is built on open APIs and supports the easy integration of enterprise solutions using familiar tools.

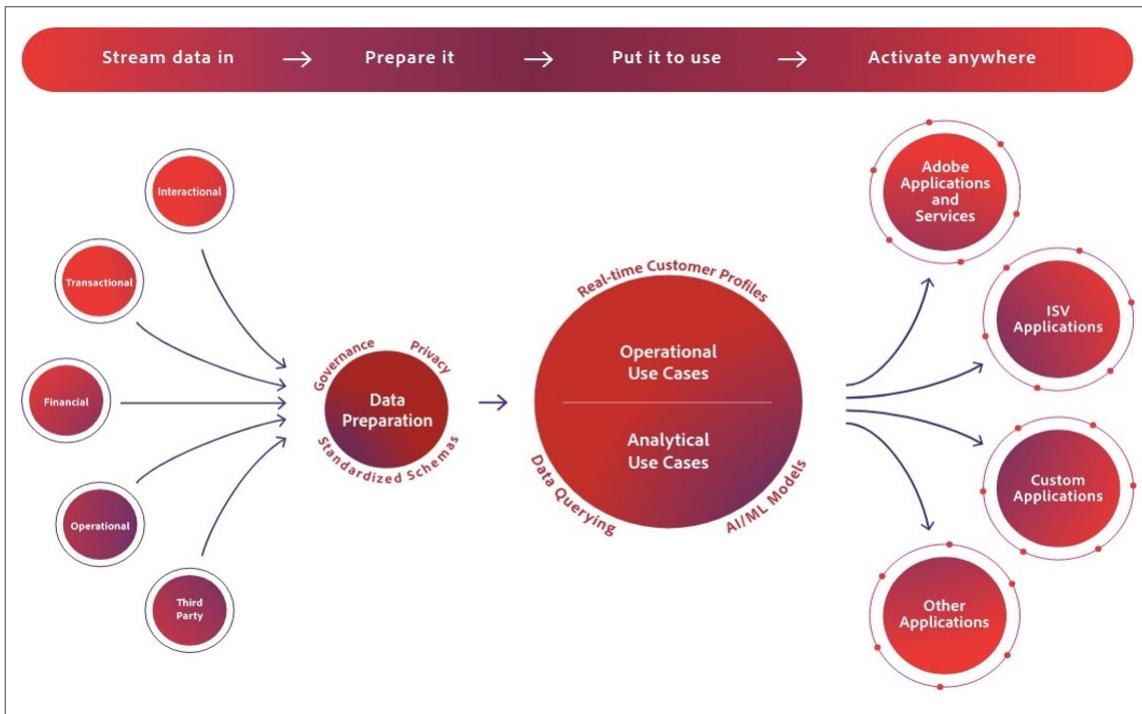
Experience Platform consists of:

- Data Foundation: Enables you to manage standard schemas, ingest data, interact with datasets, apply data governance and access data
- ML: Offers predefined models and the capability to develop custom ML models for specific needs
- Audience Activation: Includes Profile, Identity, and Edge Services. These services work together to enable you to build and activate audiences based on customer profiles containing individual attributes and behaviors.

The screenshot below represents the Experience Platform architecture:



The diagram below depicts the Experience Platform workflow:

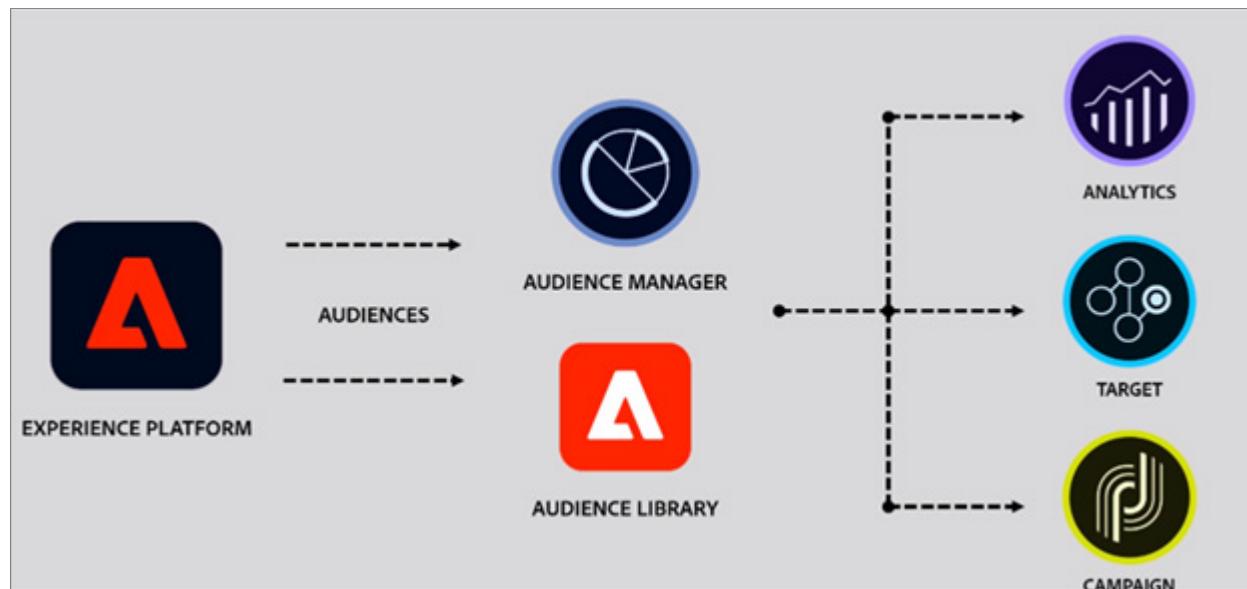


# Adobe Experience Platform and Adobe Cloud Applications Integration

You might already be using Experience Cloud applications for your marketing, advertising, analytics, and commerce needs. Experience Platform adds value by natively integrating with Experience Cloud applications. Experience Platform combines all data from all your systems, including non Adobe systems, so that you can find new insights and create real-time customer profiles.

For example, data from Adobe Analytics can be brought into Experience Platform. Every time a data point is captured on Edge, the data is available with minimal latency in Experience Platform for your further use. Data coming from Adobe Target about decisions made and content presented is available immediately in Experience Platform. Adobe Campaign Standard sends its profile and event data to Platform. Audience Manager sends its traits and audiences. With Experience Manager, you can personalize experiences in Sites and Forms. Data Collection can be used to stream data into platform using the Web SDK.

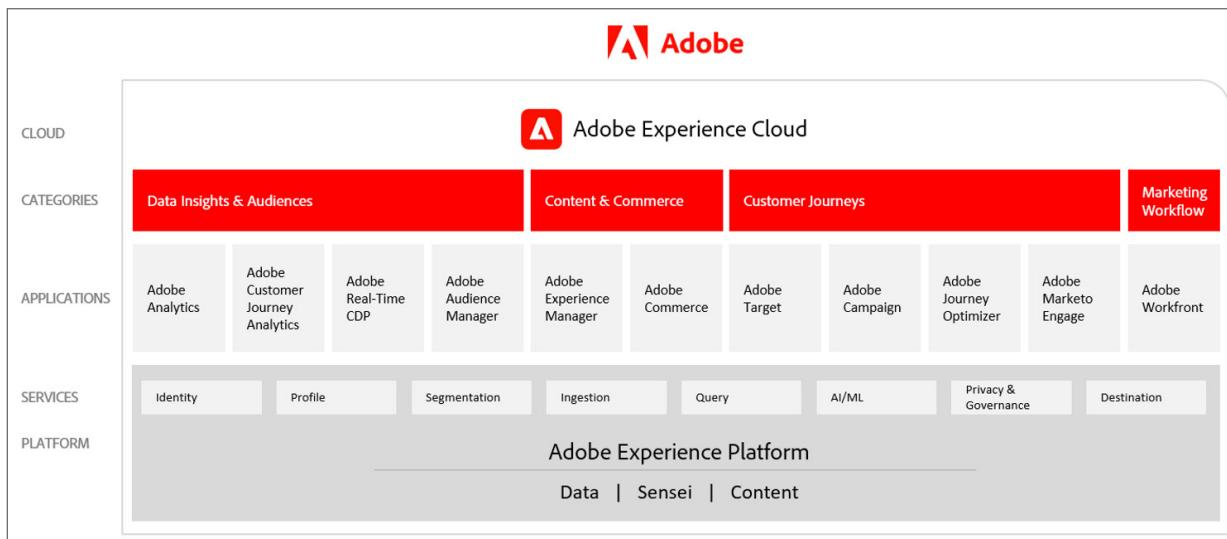
Alternatively, all the Experience Cloud applications can utilize audiences from Platform. The data will be available in your Audience Library. The diagram below illustrates the same.



The table below summarizes the benefits of Experience Platform and Experience Cloud integration:

| AAM   | Adobe Campaign  | Adobe Target  | AEM  | Adobe Analytics  |
|---|---|---|--|--|
| Supports people-based destinations to activate segments against Facebook and other applications | Utilizes audiences from Experience Platform for email campaign execution in Campaign Standard | Accesses real-time customer profile data to drive decisions and personalization | Enables segment and user level personalization within Experience Manager Sites and Forms | Leverages Analysis Workspace for analysis and visualization of data within Experience Platform |
| Shares audiences between Experience Platform and AAM  | Orchestrates journeys driven by real-time customer profile                                    |   |  |  |

The following image illustrates the Experience Cloud Application Stack:



# Application Services Powered by Experience Platform: An Overview

Application Services powered by Experience Platform provide nontechnical users, such as business analyst, marketers, and decision makers, with easy-to-use tools to address business problems without enhancing the Adobe Experience Cloud applications.

The Application Services powered by Experience Platform include:

- Real-time Customer Data Platform: Leverages the Real-time Customer Profile and segmentation capabilities to activate customer data in the form of audiences to known personally identifiable information (PII) destinations.
- Customer Journey Orchestration: Creates individual journeys for every customer based on their previous interactions and orchestrates triggered interactions, such as cart-abandonment campaigns, registration confirmations, or even location-based mobile messages.
- Customer Journey Analytics: Uses the power of Analysis Workspace to analyze the data in Experience Platform. You can bring customer data from offline or online channels into Experience Platform and break down, filter, query, and visualize the data.
- Offer Decisioning: Enables you to deliver the best offers and experiences to your customers across all touch points at the right time.

## Exercise 1: Navigate through the Experience Platform UI

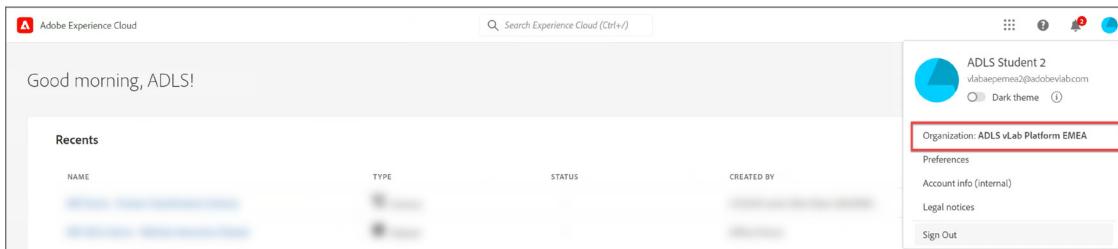
---

This exercise includes the following tasks:

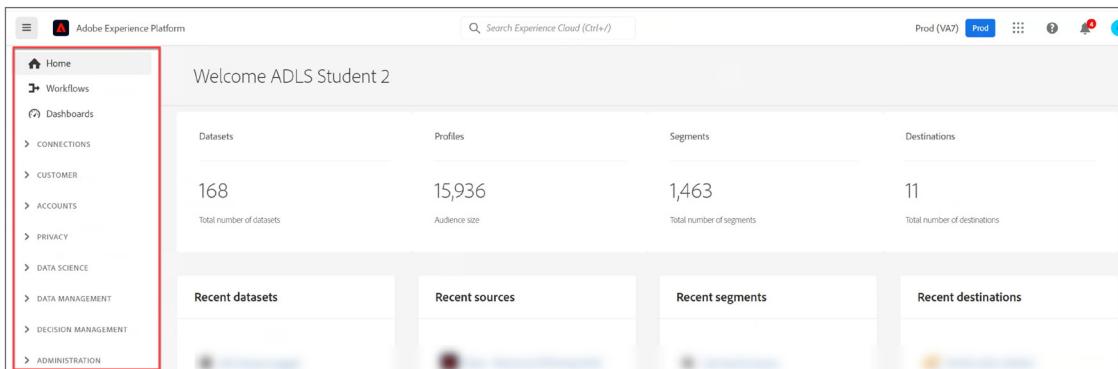
1. Log on to Experience Platform
2. Navigate through the Workflows workspace
3. Navigate through the Connections workspace
4. Navigate through the Customer workspace
5. Navigate through the Accounts workspace
6. Navigate through the Privacy workspace
7. Navigate through the Data science workspace
8. Navigate through the Data management workspace
9. Navigate through the Decision Management workspace
10. Navigate through the Administration workspace

## Task 1: Log on to Experience Platform

1. Open a web browser and type <https://platform.adobe.com>. The **Sign in** page opens.
2. Type the credentials provided to you by your instructor to sign in. The **Adobe Experience Cloud** page opens.
3. Verify that the correct instance of Experience Platform is selected for your training by clicking the profile icon on the upper-right corner. In this example, **ADLS vLab Platform EMEA** is selected, as shown:



4. (Optional) In the **Quick access** section, click **Experience Platform**. The **Adobe Experience Platform Home** page opens. The UI is segregated into different workspaces on the left pane, as shown:



## Task 2: Navigate to the Workflows workspace

- On the left pane, click **Workflows**. The **Workflows** workspace opens with the following two sections, as shown:

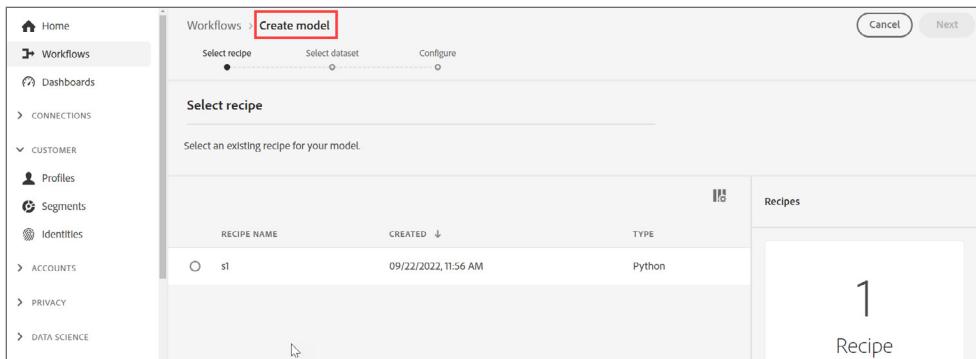
- > **Data ingestion**
- > **Models**

The screenshot shows the 'Workflows' workspace with the 'Data ingestion' section highlighted by a red box. The 'Data ingestion' section contains three cards: 'Map CSV to XDM schema', 'Create dataset from schema', and 'Create dataset from CSV file'. Below this is the 'Models' section, which contains three cards: 'Import recipe', 'Create model', and 'Publish a model as a service'.

- Click **Create dataset from schema** in the **Data ingestion** section. The **Launch** button is displayed on the right pane.
- Click **Launch**. The **Create dataset from schema** page opens, as shown, with the existing schemas. On this page, you can select an existing schema and associate it with a new dataset.

The screenshot shows the 'Create dataset from schema' page. The 'Select schema' step is selected. It displays a table of existing schemas with columns: NAME, ENABLED FOR PROFILE, CLASS, and BEHAV. Two schemas are listed: 'Ademola Website Registration Schema' (Enabled, XDM Individual Profile, Recor) and 'AEP Demo - Luma Telco Call Center Interacti' (Not enabled, XDM ExperienceEvent, Time). A 'schemas' sidebar on the right lists 'Most recently created' schemas, including 'Custom XDM mixin for...' (09/26/2022).

4. Click **Cancel** at the upper-right corner to return to the **Workflows** workspace.
5. Click **Create model** in the **Models** section. The **Launch** button is displayed on the right pane.
6. Click **Launch**. The **Create model** page opens, as shown. On this page, you can either select an existing machine learning recipe or import one to represent your business model.



7. Click **Cancel** to return to the **Workflows** workspace.

### Task 3: Navigate through the Connections workspace

1. On the left pane, expand **CONNECTIONS**. The **CONNECTIONS** workspace displays the following two components:
  - › **Sources**
  - › **Destinations**
2. Click **Sources**. The **Sources** page opens.
3. On the **Catalog** tab, scroll down to the **CATEGORIES** section and notice the source connectors under it such as **Adobe applications**, **Advertising**, **Analytics**, **Cloud storage**, and **Consent & Preferences**, as shown:

The screenshot shows the Adobe Experience Platform Connections workspace. The left sidebar has several sections: Home, Workflows, Dashboards, CONNECTIONS (with Sources and Destinations), CUSTOMER (Profiles, Segments, Identities), ACCOUNTS (Profiles), PRIVACY (Policies), Requests, and Audits. The CONNECTIONS section is expanded, and SOURCES is selected. The main area has tabs: Catalog (which is selected), Dataflows, Accounts, and System View. Below the tabs is a banner: "Enable your data for use in Real-time Customer Data Platform" with a "Introduction to sources" link. To the right of the banner are filter options: "All sources" (selected) and "My sources". Below that is a search bar with a magnifying glass icon and the word "Search". Under the search bar is a section titled "Adobe applications" with a list of connectors: Adobe Analytics, Advertising, Analytics, Cloud storage, and Consent & Preferences. The "Adobe applications" section is highlighted with a red box. To the right of the connectors are "Set up" and "..." buttons. The overall interface is clean and modern, designed for managing data connections in the platform.

4. Click **Destinations** on the left pane. The **Destinations** page opens.

5. On the Catalog tab, under CATEGORIES, notice destinations such as **Adobe applications**, **Advertising**, **Cloud storage**, **CRM**, and **Email Marketing**, as shown:

The screenshot shows the Destinations Catalog interface. The left sidebar includes Home, Workflows, Dashboards, CONNECTIONS (with Sources and Destinations selected), CUSTOMER (Profiles, Segments, Identities), ACCOUNTS (Profiles), and PRIVACY. The main area has tabs for Overview, Catalog (selected), Browse, Accounts, and System View. Under Catalog, 'All destinations' is selected over 'My destinations'. Under TYPES, 'Connections' is checked over 'Extensions'. Under CATEGORIES, a red box highlights 'Adobe applications', 'Advertising', 'Cloud storage', 'CRM', and 'Email marketing'. A connection for '(V1) Marketo Engage Connection' is shown with a 'Set up' button. Below it is a section for 'Advertising' with a connection for 'Advertising Cloud DSP Connection'.

## Task 4: Navigate through the Customer workspace

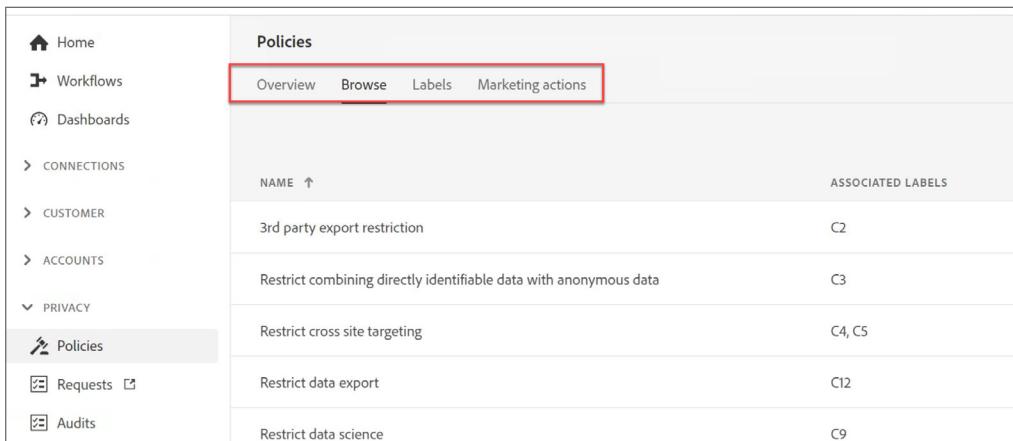
1. On the left pane, expand **CUSTOMER**. The **CUSTOMER** workspace displays the following three components:
  - › **Profiles**
  - › **Segments**
  - › **Identities**
2. Click **Profiles**. The **Profiles** page opens with the following tabs:
  - › **Overview**: Provides you with learning resources.
  - › **Browse**: Enables you to search and select a profile by using an identity namespace and an identity value.
  - › **Merge Policies**: Helps you to create merge policies.
  - › **Union Schema**: Enables you to view the union schemas. A union schema is comprised of the schemas of the same class.
3. On the left pane, click **Segments**. The **Segments** page opens with the **Overview**, **Browse**, and **Feeds** tabs. On this page, you can explore the available learning resources, create new segments, browse through the existing segments, or click an existing segment to analyze or edit it.
4. Click **Identities**. The **Identities** page opens with the **Overview**, **Browse**, and **Identity Graph** tabs. On this page, you can explore the available learning resources, create a new identity namespace, and browse or search for existing identities.

## Task 5: Navigate through the Accounts workspace

1. On the left pane, expand **ACCOUNTS**. The **ACCOUNTS** workspace includes the **Profiles** option.
2. Click **Profiles**. The **Account Profiles** page opens.
3. In the **Overview** tab, you can add widgets, modify dashboard, and view predictive scoring distribution and predictive scoring top influential factors.
4. On the **Browse** tab, you can browse account profiles by selecting a source system and entering an account ID.

## Task 6: Navigate through the Privacy workspace

1. On the left pane, expand **PRIVACY**. The **PRIVACY** workspace displays the following three components:
  - › Policies
  - › Requests
  - › Audits
2. Click **Policies**. The **Policies** page opens with the **Overview**, **Browse**, **Labels**, and **Marketing actions** tabs, as shown:

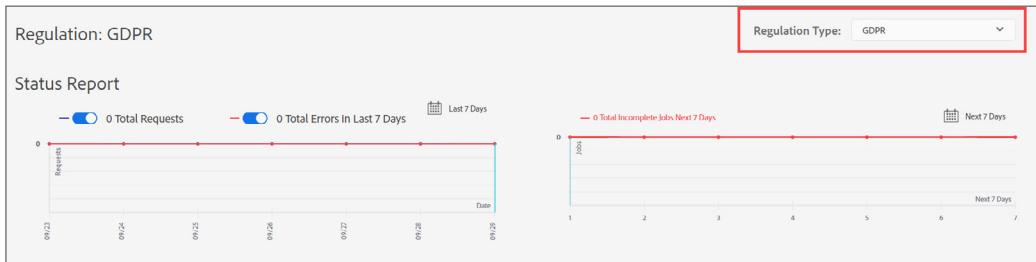


The screenshot shows the Adobe Experience Platform interface. On the left, there's a navigation sidebar with links like Home, Workflows, Dashboards, CONNECTIONS, CUSTOMER, ACCOUNTS, and PRIVACY. Under PRIVACY, the Policies link is highlighted. The main content area has a header 'Policies' with tabs: Overview, Browse, Labels, and Marketing actions. The Marketing actions tab is currently selected, indicated by a red border around its tab. Below the tabs is a table with columns 'NAME' and 'ASSOCIATED LABELS'. There are four rows in the table:

| NAME  | ASSOCIATED LABELS |
|---|-------------------|
| 3rd party export restriction                                      | C2                |
| Restrict combining directly identifiable data with anonymous data | C3                |
| Restrict cross site targeting                                     | C4, C5            |
| Restrict data export  | C12               |
| Restrict data science   | C9                |

- › **Overview**: Provides you with learning resources.
- › **Browse**: Enables you to search and display the privacy policies, enable or disable individual privacy policies, and create new privacy policies.
- › **Labels**: Enables you to find each label and its policy description. You can also create new labels.
- › **Marketing actions**: Enables you to review the existing marketing actions and create new ones.

The **Requests** section under **PRIVACY** enables you to create and manage privacy requests. You must be allowlisted to access the **Privacy Service** interface. The **Privacy Service** interface includes two sections, **Status Report** and **Job Requests**. These sections enable you to view the status of the privacy jobs and the current selected regulation for the displayed jobs. In this example, the selected regulation type is GDPR, as shown:



The **Privacy Service** interface enables you to:

- > Create new privacy job requests for the following three regulation types:
    - » The European Union's General Data Protection Regulation (GDPR)
    - » The California Consumer Privacy Act (CCPA)
    - » Thailand's Personal Data Protection Act (PDPA\_THA)
  - > Track the details of previously submitted jobs
3. On the left pane, click **Audits**. The activity log is displayed.

## Task 7: Navigate through the Data Science workspace

1. On the left pane, expand **DATA SCIENCE**. The **DATA SCIENCE** workspace contains the following three components:
  - › **Notebooks**
  - › **Models**
  - › **Services**
2. Click **Notebooks**. The **Notebooks** page opens with the **Overview** and **JupyterLab** tabs, as shown:

The screenshot shows the left sidebar with the 'DATA SCIENCE' section expanded, revealing 'Notebooks', 'Models', and 'Services'. The main area is titled 'Notebooks' and displays two tabs: 'Overview' and 'JupyterLab'. A red box highlights the 'Overview' tab. Below the tabs, a section titled 'Getting started with Notebooks' features a lightbulb icon and a 'Explore Notebooks' button. At the bottom, there's a brief description of JupyterLab: 'JupyterLab is tightly integrated into Experience Platform. It provides an interactive development environment to explore, analyze, and model your data.'

3. On the left pane, click **Models**. The **Models** page opens. The page contains the following tabs:
  - › **Overview**: Enables you to explore learning resources
  - › **Browse**: Enables you to create models or browse the existing models
  - › **Recipes**: Enables you to import recipes or browse the existing recipes

4. On the left pane, click **Services**. The **Services** page opens, as shown. The **Services** page provides resources on Adobe Intelligent Services.

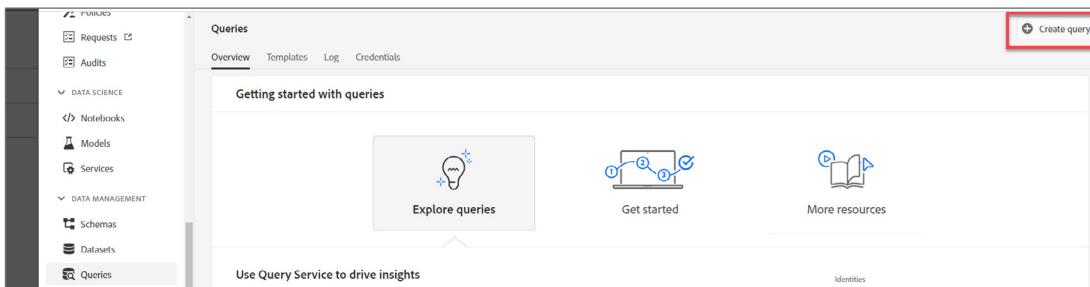
The screenshot shows the Adobe Experience Platform interface. On the left, a vertical navigation bar lists various services: Home, Workflows, Dashboards, CONNECTIONS, CUSTOMER, ACCOUNTS, PRIVACY, DATA SCIENCE, Notebooks, Models, and Services. The 'Services' option is selected, indicated by a highlighted background. The main content area is titled 'Services' and has a sub-section titled 'Browse'. A prominent heading says 'Leverage the power of AI and ML services.' Below this, a text block explains that Adobe Experience Platform provides a framework to operationalize AI and ML services, mentioning Adobe's Intelligent Services. Two buttons are present: 'Learn about Adobe services' and 'Learn how to publish services'. At the bottom, there is a section titled 'Adobe services' with a small icon and some decorative graphics.

## Task 8: Navigate through the Data Management workspace

- On the left pane, click **DATA MANAGEMENT**. The following five components appear:
  - › **Schemas**
  - › **Places**
  - › **Datasets**
  - › **Queries**
  - › **Monitoring**
- Click **Schemas**. The **Schemas** page opens with the **Overview**, **Browse**, **Classes**, **Field groups**, and **Data types** tabs, as shown. On the **Schemas** page, you can explore the available learning resources, create new schemas, and browse or search for the existing schemas.

- On the **Browse** tab, click a schema to view its composition, structure, and schema properties.
- On the left pane, click **Places**. The **POI Management** page opens. In this page, you can create and import Point of Interests (POIs). A POI is a geo location that is of interest to your organization.
- On the left pane, click **Datasets**. The **Datasets** page opens with the **Overview** and **Browse** tabs. On the **Datasets** page, you can explore the available learning resources, create new datasets, and browse or search for the existing datasets.
- On the **Browse** tab, click a dataset to analyze the data on the **Dataset activity** and **Data governance** tabs of the schema.

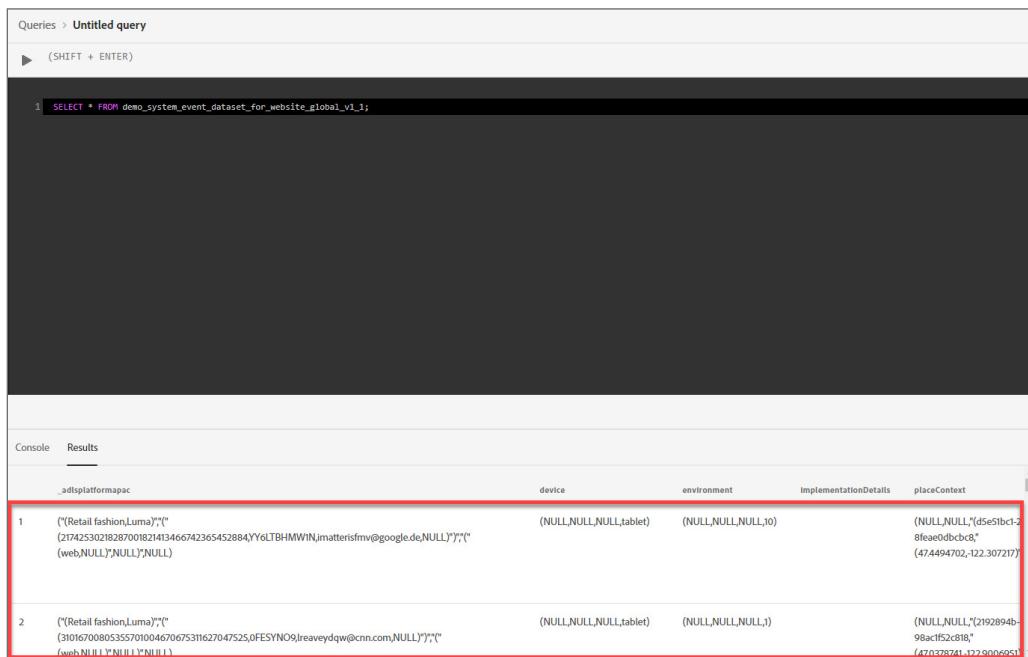
7. On the left pane click **Queries**. The **Queries** page opens with the **Overview**, **Templates**, **Log**, and **Credentials** tabs. On the **Queries** page, you can explore the available learning resources, create queries, browse or search for the existing connections, and view the Postgres credentials.
8. At the upper-right corner, click **Create query**, as shown. The **Untitled query** page opens as the query editor. You can write, validate, and run queries for customer experience data within the Experience Platform UI by using the query editor. A blue warning pop-up message, indicating that you understand the limitations and obligations of the data may appear.



9. Select the **Don't show again** checkbox and click the close (x) button in the pop-up message to ensure the message does not appear again.
10. Click the first line in the query editor and type the **SELECT\*FROM demo\_system\_event\_dataset\_for\_website\_global\_v1\_1;** SQL command, as shown. Alternatively, you can also copy the command from the **Exercise Files > Your region folder > Your region CodeFile > CodeFile\_M1\_Ex1** file from the exercise folder provided to you by your Instructor. For example, **Exercise Files > APAC > APAC CodeFile > CodeFile\_M1\_Ex1**.

```
1 | SELECT * FROM demo_system_event_dataset_for_website_global_v1_1;
```

11. Click the play button or press the Shift+Enter keys on your keyboard to execute the query. The **Results** tab displays the result of the query, as shown. The **Console** tab displays the status of the query execution.



The screenshot shows the 'Queries > Untitled query' page. At the top, there's a play button labeled '(SHIFT + ENTER)'. Below it, the query code is displayed:

```
1 SELECT * FROM demo_system_event_dataset_for_website_global_v1_1;
```

The results are shown in a table with two rows. The columns are: adspiplatformpac, device, environment, ImplementationDetails, and placeContext. The first row has a red border around its entire content. The second row also has a red border around its entire content.

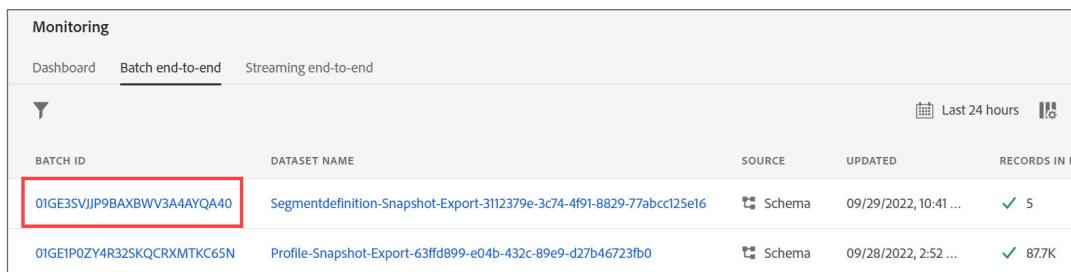
| adspiplatformpac   | device                  | environment         | ImplementationDetails                                       | placeContext |
|--|-------------------------|---------------------|---|--------------|
| 1 ("(Retail fashion,Luma)";"("217425302892870018241546742365452884YY6LTBHMWIN,imatterisfmv@google.de,NULL)");"("web,NULL");NULL);NULL) | (NULL,NULL,NULL,tablet) | (NULL,NULL,NULL,10) | (NULL,NULL,"(d5e51bc1-8feae0dbcb0,"(474494702,-122.307217)) |              |
| 2 ("(Retail fashion,Luma)";"("31016700805355701004670675311627047525,0FESYN09,leaveydqw@cnn.com,NULL)");"("web,NULL");NULL);NULL)      | (NULL,NULL,NULL,tablet) | (NULL,NULL,NULL,1)  | (NULL,NULL,"(2192894b-98acf52c818,"(470378741,122.9006951)) |              |

12. Click **Cancel** at the upper-right corner to return to the **Queries** page.
13. On the left pane, click **Monitoring**. The **Monitoring** page opens with the **Dashboard**, **Batch end-to-end**, and **Streaming end-to-end** tabs.
14. Click the **Batch end-to-end** tab. There are two types of batches, successful and failed. The successful batches contain the records that were persisted successfully into Data Lake. Failed batches contain the records that encountered failures of the ingestion path to Experience Platform.



**Note:** If you do not see any batches, click the calendar icon at the upper-right corner and change the date range. Try different date ranges. Successful batches have a green check mark in the **RECORDS IN DATASET** column along with the corresponding number of records. Failed batches have a red triangular icon with the hyperlinked text displayed as **Failures**.

15. Click a successful entry in the **BATCH ID** column, as shown. The **Batch overview** tab opens with the total number of records ingested, along with other information about the batch.



| BATCH ID                   | DATASET NAME   | SOURCE | UPDATED               | RECORDS IN D |
|----------------------------|--|--------|-----------------------|--------------|
| 01GE3SVJJP9BAXBWV3A4AYQA40 | Segmentdefinition-Snapshot-Export-3112379e-3c74-4f91-8829-77abcc125e16 | Schema | 09/29/2022, 10:41 ... | ✓ 5          |
| 01GE1P0ZY4R32SKQCRXMTKC65N | Profile-Snapshot-Export-63ffd899-e04b-432c-89e9-d27b46723fb0           | Schema | 09/28/2022, 2:52 ...  | ✓ 87.7K      |

16. On the left pane, click **Monitoring** to return to the **Monitoring** page.
17. Click a failed **BATCH ID**. The **Batch overview** tab opens. The tab displays the details including batch ID, record count, error code, and the reason for failure.

 **Note:** If you are unable to view any batch ID that is failed, click the calendar icon at the upper-right corner and select a date range, such as **Last 30 days**, and click **Apply**.

18. On the left pane, click **Monitoring** to return to the **Monitoring** page.
19. Click the **Streaming end-to-end** tab. The **Streaming end-to-end** section displays the **Ingestion rate** section and details of the ingestion. The **Ingestion rate** section displays the rate of messages being streamed into Experience Platform. By default, the section displays the trend for the last seven days, but you can increase or decrease the range. The details section provides a detailed view of how the streamed data is getting persisted into Data Lake.

## Task 9: Navigate through the Decision Management workspace

- On the left pane, expand **DECISION MANAGEMENT**. The following components are available:
  - Offers
  - Components
- Click **Offers**. The **Offers** page opens. This page includes **Overview**, **Offers**, **Collections**, **Decisions**, and **Simulation** tabs, as shown:

| Offers                                    | Offer collections  | Offer decisions                       |
|---|--------------------|---------------------------------------|
| Luma - 5% discount on Women's Cat...      | Luma - Mobile App  | Luma - 5% Discount on Men/Women ...   |
| Luma - Deals - Fallback                   | Luma - Loyalty     | Luma - Loyalty Status/Upgrade Camp... |
| Luma - Free Delivery on next order (v...  | Luma               | Luma - Call Center Decisions          |
| Luma - Join Loyalty Club (v1.2)           | Luma - Deal Offers | Luma - Use Mobile App Campaign        |
| Luma - Latest Offers Available in Mobi... |                    | Luma - Deals                          |

- Overview:** Explore learning resources
  - Offers:** Access your personalized and fallback offers or create new ones
  - Collections:** Create and manage static and dynamic offer collections
  - Decisions:** Create and manage decisions to deliver your offers
  - Batch decisions:** Create job requests to deliver offer decisions to all profiles in a given Adobe Experience Platform segment
  - Simulation:** Add decision scope, simulate offer decisions, and view the results of offer decision for a selected profile
- On the **Offers** tab, click **Create offer**. The **New offer** dialog box opens with the option to select an offer type, as shown. The following offer types are available:
    - Personalized offer:** Is a customizable message based on eligibility rules and constraints
    - Fallback offer:** Is the default offer displayed when an end user is not eligible for personalized offers

| NAME                                  | TYPE         | DATE   | END DATE   | LAST MODIFIED        |
|---------------------------------------|--------------|--------|------------|----------------------|
| 15% Off Winter Sale                   | Personalized | 1/2022 | 12/31/2022 | 04/14/2022, 2:31 PM  |
| AEP ADLS DEMO - Purchase              | Personalized | 1/2021 | 12/31/2021 | 01/12/2022, 2:13 PM  |
| ADLS AEP DEMO - Not Purchase Fallback | Fallback     | -      | -          | 01/12/2022, 2:08 PM  |
| ADLS DEMO - Discover AEP              | Fallback     | -      | -          | 03/29/2021, 11:04 AM |
| ADLS DEMO - Training Offer            | Personalized | 5/2021 | 07/30/2021 | 03/29/2021, 11:03 AM |
| ADLS DEMO - Adobe Offer               | Personalized | 8/2021 | 07/30/2021 | 03/29/2021, 11:02 AM |

4. Click **Cancel** in the **New Offer** dialog box.
5. On the left pane, click **Components**. The **Components** page opens. This page includes **Placements**, **Tags**, **Rules**, and **Ranking** tabs, as shown:

| Components |                    |              |              |                      |
|------------|--------------------|--------------|--------------|----------------------|
|            | NAME               | CHANNEL TYPE | CONTENT TYPE | LAST MODIFIED        |
|            | Matthias           | Web          | HTML         | 09/10/2021, 12:30 PM |
|            | SMS - JSON         | Mobile       | JSON         | 03/29/2021, 8:57 AM  |
|            | Non-digital - Text | Non-digital  | Text         | 03/29/2021, 8:56 AM  |
|            | Email - Image      | Email        | Image        | 03/29/2021, 8:56 AM  |
|            | Email - Text       | Email        | Text         | 03/29/2021, 8:56 AM  |
|            | Email - JSON       | Email        | JSON         | 03/29/2021, 8:56 AM  |
|            | Email - HTML       | Email        | HTML         | 03/29/2021, 8:55 AM  |

- › **Placements**: Create and manage placements where your offers are displayed.
  - › **Tags**: Create and manage tags to organize and filter your offers.
  - › **Rules**: Manage the conditions under which your offers are presented.
  - › **Rankings**: Create and manage ranking formulas to determine which offer should be presented first for a given placement.
6. Click **Create placement** at the upper-right corner. The **Create placement** dialog box opens with fields to update name, channel type, content type, and description.
  7. Click **Cancel** in the **Create placement** dialog box.

## Task 10: Navigate through the Administration workspace

1. On the left pane, expand **ADMINISTRATION**. The following components are available:
  - › **Alerts**
  - › **Sandboxes**
  - › **License Usage**
2. Click **Alerts**. The **Alerts** page opens. This page includes two tabs—**Browse** and **History**. The **Browse** tab lists the available rules that may trigger an alert. The **History** tab displays the history of received alerts for your organization, including the rule that triggered the alert, triggered date, and resolved date.

---

 **Note:** Experience Platform provides virtual sandboxes that partition a single Experience Platform instance into separate, isolated virtual environments. If you have the required permissions, you will see the **Sandboxes** option under **ADMINISTRATION**. The **Sandboxes** page includes two tabs, **Overview** and **Browse**. The **Overview** tab includes learning resources. The **Browse** tab includes a list of sandboxes. On the **Browse** tab, users with required permissions can create new sandboxes.

---

3. On the left pane, click **License Usage**. The **License Usage** page opens. This is a dashboard that displays an overview of your organization's license-related data for Experience Platform.
4. On the left pane, click **Home** to return to the home page.

## Module 2

# Managing Data in Adobe Experience Platform

---

### Introduction

Customer data is crucial for any experience business. It is important to manage the customer data in real time to provide personalized and meaningful experiences to customers.

Adobe Experience Platform helps centralize and standardize customer data to create real-time customer profile. You can analyze and prepare the data and ensure that the organizational policies are implemented when using the data.

### Objectives

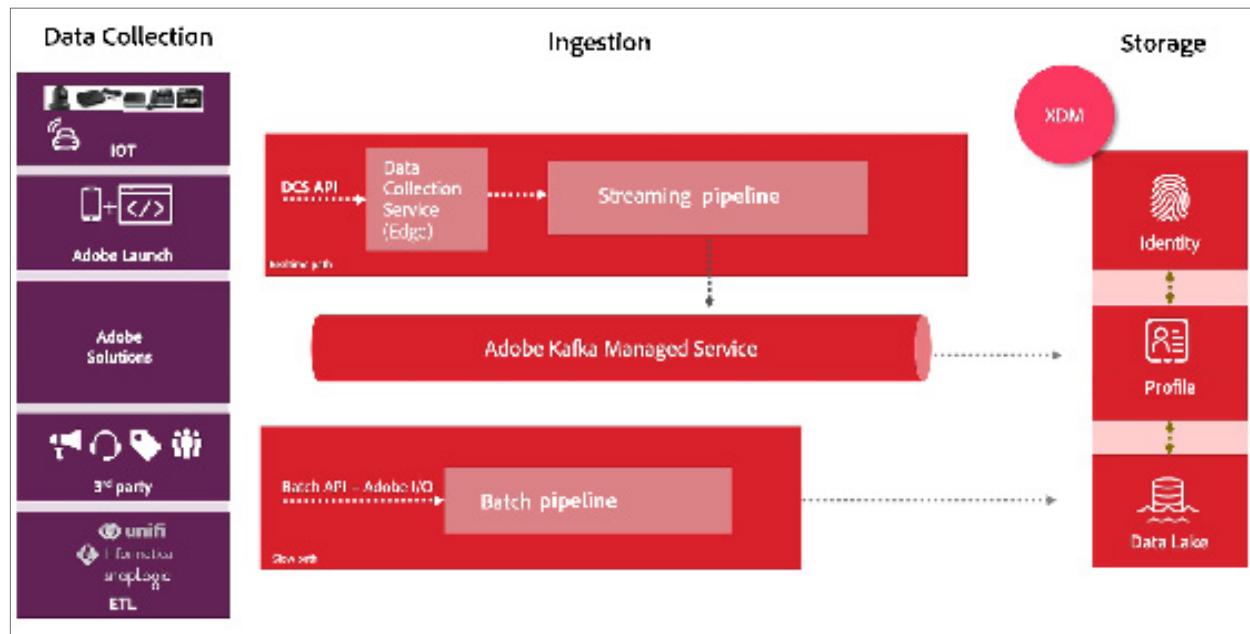
After completing this module, you will be able to:

- Explain the data ingestion pipeline in Experience Platform
- Explain the data ingestion methods in Experience Platform
- Explain Experience Data Model (XDM)
- Explain data governance and privacy in Experience Platform
- Verify data ingestion from a website into Experience Platform
- Ingest data by using a comma-separated values (CSV) file mapped to the XDM schema
- Apply data governance to a dataset

# Data Ingestion Pipeline in Experience Platform

Experience Platform enables you to ingest data from external sources. This data can be stored and used to create meaningful customer experiences.

The following diagram displays the data ingestion pipeline in Experience Platform:



The data ingestion pipeline includes:

## Data collection

- Data is collected from various sources such as Adobe applications, enterprise sources, application programming interface (APIs), and web or mobile software development kit (SDKs). The data is ingested for data preparation and storage.

## Ingestion

- **Batch pipeline:** Ingests large volumes of data. The pipeline contains batch ingestion APIs/SDKs, the connectors for non-Adobe applications, bidirectional integration with ETL vendors to access 100+ applications, and batch data access APIs.

- Streaming pipeline: Ingests real-time data and activates the customer data and insights in real time. The pipeline contains events at the edge, streams APIs to process collected data, and integrates with the Adobe I/O events to publish events back to Adobe and non-Adobe applications.
- XDM: Standardizes the structure of the ingested data and simplifies the process of gaining insights from massive amount of data.

## Storage

- Profile: Provides a real-time view of customer data. You can use this data to create segments and audiences that can be activated within Adobe and non-Adobe applications.
- Identity Service: Stitches user identities together across devices and systems for a complete view within real-time customer profile.
- Data Lake: Stores customer data in raw format.

# Data Ingestion Methods in Experience Platform

---

The data ingestion methods in Experience Platform are:

- Batch ingestion
- Streaming ingestion

## **Batch Ingestion**

Batch ingestion enables you to ingest data into Experience Platform as batch files. Batches are units of data that consist of one or more files to be ingested as a single unit. These batches provide metadata that describes the number of records successfully ingested, and any failed records and associated error messages. This method should be used to ingest data files such as flat CSV files and Parquet data frames that are manually uploaded.

## **Streaming Ingestion**

Streaming ingestion enables you to send data from the client-side and server-side devices to Experience Platform in real time. The data is streamed by using the data inlets into Experience. The data is persisted in streaming-enabled datasets within the Data Lake.

You can use source connectors to ingest data from external source into Experience Platform by using the batch ingestion or streaming ingestion methods.

## XDM: An Overview

---

XDM is a foundational framework in Experience Platform and enables you to:

- Represent customer data in a single standard data model
- Use common structures and definitions to communicate with the Experience Platform services
- Reuse data and easily bring third-party data into Experience Platform

The schemas in XDM enable you to define the structure of the data in a consistent and reusable way.

You can:

- Represent and validate the structure and format of data
- Apply constraints and expectations to data for validation
- Interpret data consistently regardless of its origin, and remove the need for translation across applications

For example, schemas provide an abstract definition of a real-world object, such as a person, and the data associated with that object, such as first name, last name, or birthday.

Schemas describe data that can be categorized as:

- Record data: Provides information about the attributes of a subject, such as city, location, name, level, or loyalty points. A subject could be an organization or an individual.
- Time series data: Provides a snapshot of the system at the time of an action, such as a click, purchase, or cart abandonment, taken either directly or indirectly by a subject.

Schemas contain a class but may or may not contain field groups:

- Class: Defines the attribute or behavioral aspects of the record or time series data in schema. The two types of classes in Experience Platform are:
  - XDM Individual Profile: Standard record-based class that captures the attributes and interests of both identified and partially identified individuals. For example, name, date of birth, location, and email address are XDM Individual Profile classes.
  - XDM ExperienceEvent: Standard timeseries-based class that captures the timestamp when a specific event occurs or a certain set of conditions have been reached. For example, id, event id, and event type are XDM ExperienceEvent classes.

- Schema field group: Defines one or more fields that can be included in a schema. For example, personal details, hotel preferences, or address are schema field groups. A schema's class determines the field groups that are eligible to be used in the schema.

# Data Governance and Privacy Service in Experience Platform

## Data Governance

Experience Platform enables you to bring data from multiple enterprise systems together to identify, understand, and engage customers. This data can be subject to usage restrictions defined by your organization or by legal regulations. Hence, it is important to ensure that your data operations within Experience Platform are compliant with data usage policies.

Data governance enables you to:

- Ensure compliance with regulations, restrictions, and policies applicable to data usage
- Control the usage of data for marketing actions

The General Data Protection Regulation (GDPR), California Consumer Privacy Act (CCPA) and data governance frameworks in Experience Platform are used to manage privacy regulations.

## Data Governance Framework

The data governance framework simplifies and streamlines the process of categorizing data and creating data usage policies. After data usage policies are in place, marketing actions can be evaluated to ensure the correct use of data.

The three key elements of the data governance framework are:

- Labels
- Policies
- Enforcement

## Labels

Labels enable you to categorize datasets and fields according to usage policies that apply to the data. You can apply the labels at any time. The best practice is to label data immediately after the data is ingested into Experience Platform.

The data governance framework includes predefined labels that can be used to categorize data in three ways:

- Identity I Data Labels: Categorize data that can identify or contact a specific person
- Contract C Data Labels: Categorize data that has contractual obligations or is related to customer data governance policies
- Sensitive S Data Labels: Categorize data related to sensitive data, such as geographic data

## Policies

Policies are rules that enable you to:

- Support data compliance
- Describe the kinds of marketing actions that you can perform or are restricted from performing on the data within Experience Platform

For example, you want to export a dataset to a third-party service, and there is a policy in place mentioning that specific types of data, such as personally identifiable information, cannot be exported and an I label (Identity data) has been applied to the dataset. In this case, you will receive a response from the Policy Service mentioning that a data usage policy is violated.

## Enforcement

After the data is labeled and usage policies are defined, you can enforce data usage compliance and prevent data operations that include policy violations. When activating audience segments to destinations in real-time customer data platform, data governance automatically enforces usage policies in case of violations.

## Privacy Service

Privacy Service provides a Representational State Transfer (RESTful) API and user interface that enables you to:

- Manage customer personal data requests
- Submit requests to access and delete personal customer data from Adobe Experience Cloud applications
- Facilitate automated compliance with legal and organizational privacy regulations

The violation of privacy regulations results in major fines and disrupted data operations for your business.

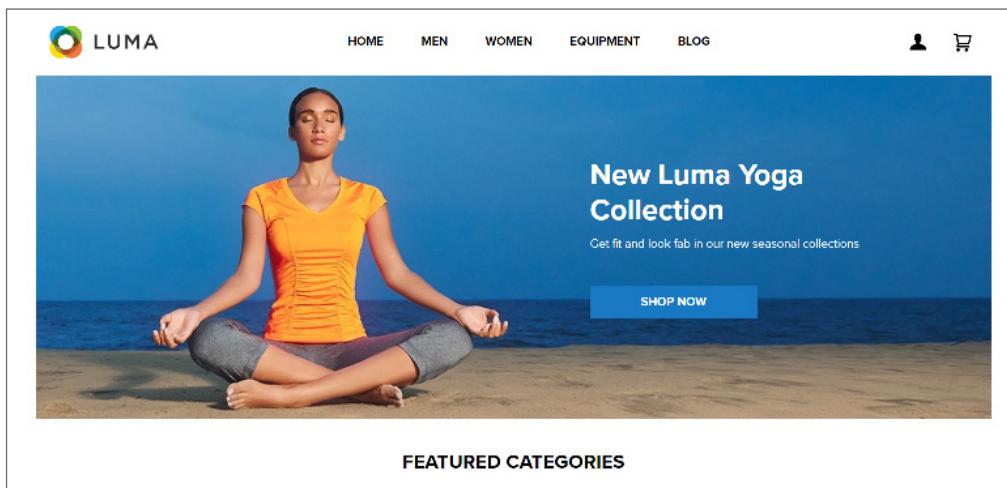
# Exercise 1: Verify data ingestion from a website into Experience Platform

This exercise includes the following tasks:

1. Generate the web traffic and analyze the calls sent to Experience Platform
2. Verify the data in Experience Platform

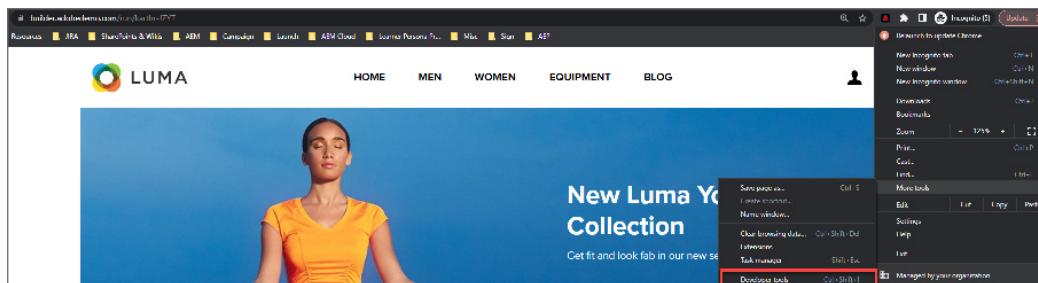
## Task 1: Generate the web traffic and analyze the calls sent to Experience Platform

1. Open an incognito window in the Google Chrome browser.
2. Copy the LUMA website URL provided in the lab URLs document, specific to your region provided by your Instructor, and paste it in the incognito browser.
3. Press the Enter key. The LUMA website opens, as shown:

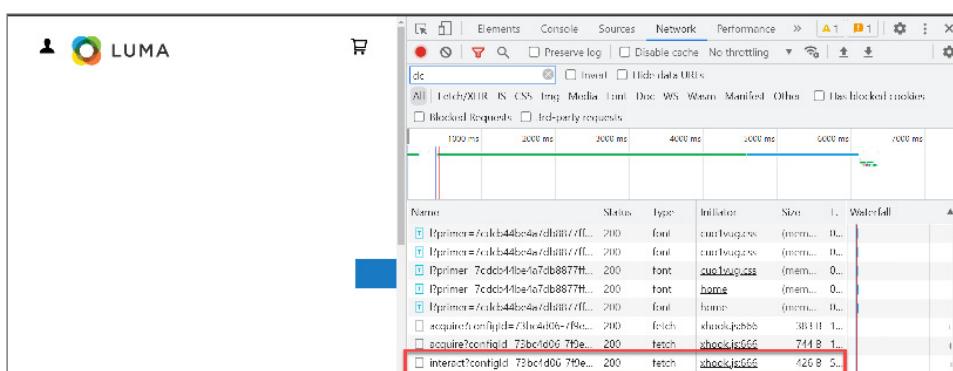


**Warning:** Do not register at this point. You will register later in the class.

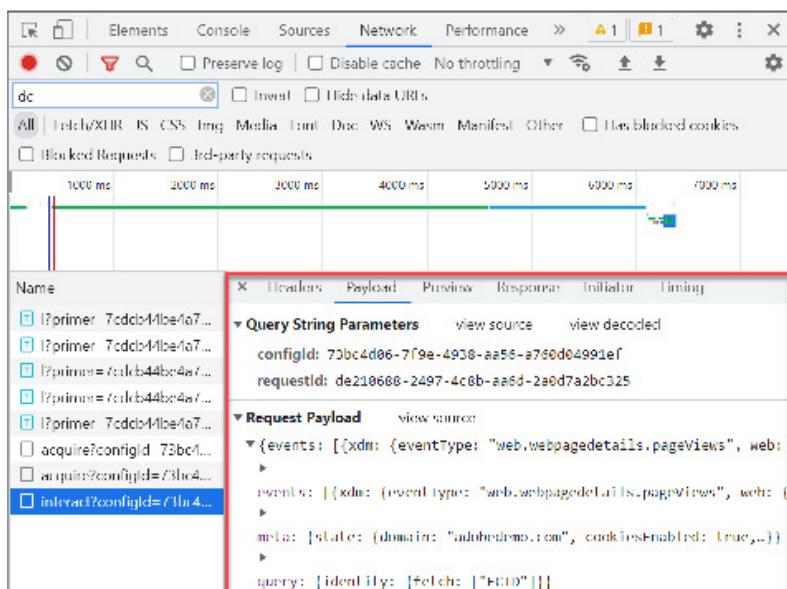
4. Click the three dots at the upper-right corner of the browser and select **More tools > Developer tools**, as shown, to open the Chrome developer tools. The developer tools section opens on the right pane on the website.



5. Click the **Network** tab to view the calls sent from the LUMA website to various servers, including the calls to the distributed control system (DCS) endpoint of Experience Platform.
6. Press the **Ctrl+R** keys to view the updated calls.
7. Type **dc** in the **Filter** box. The call to the DCS endpoint of Experience Platform is displayed, as shown:



8. Click the call to the DCS endpoint. The details of the call is displayed on the right, as shown:



9. In the Payload tab, under the Request Payload section, click view source, as shown.

The screenshot shows the Network tab with several requests listed. The 'Request Payload' section for one request is expanded, showing the raw JSON code. A red box highlights the 'view source' button next to the 'Request Payload' label.

```
{
  "events": [
    {
      "xdm": {
        "eventType": "web.webpagedetails.pageViews",
        "web": {
          "events": [
            {
              "xdm": {
                "eventType": "web.webpagedetails.pageViews",
                "web": {
                  "meta": {
                    "state": {
                      "domain": "adobedemo.com",
                      "cookiesEnabled": true
                    }
                  }
                }
              }
            }
          ]
        }
      }
    }
  ]
}
```

The complete call sent to Experience Platform is displayed, as shown:

The screenshot shows the Network tab with several requests listed. The 'Request Payload' section for one request is expanded, showing a large JSON object representing the event payload. A red box highlights the entire JSON content.

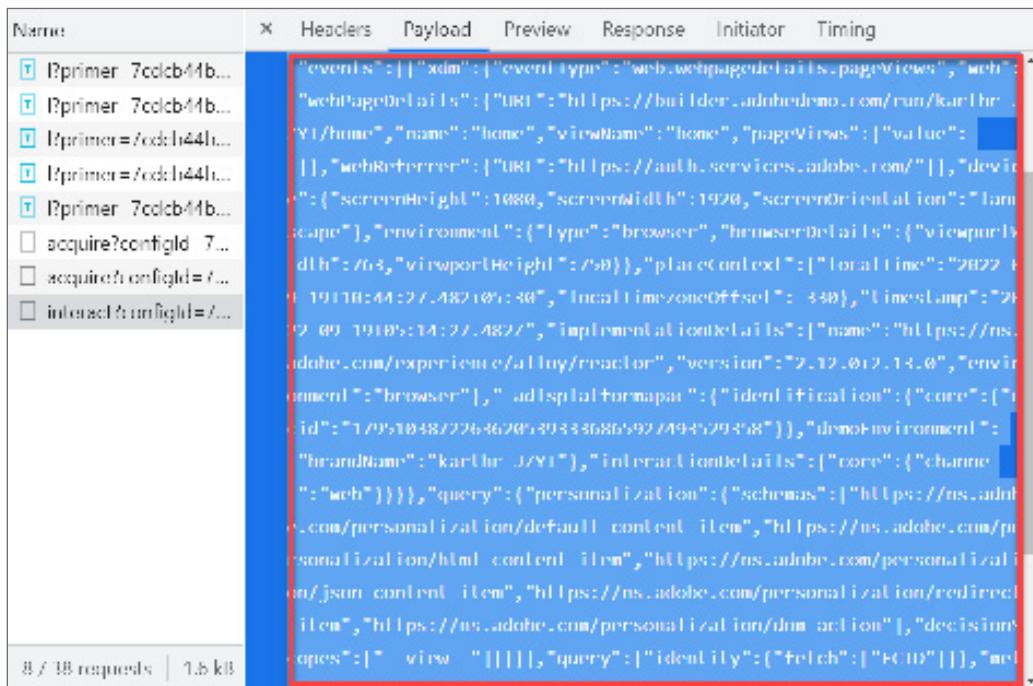
```
{
  "events": [
    {
      "xdm": {
        "eventType": "web.webpagedetails.pageViews",
        "web": {
          "events": [
            {
              "xdm": {
                "eventType": "web.webpagedetails.pageViews",
                "web": {
                  "meta": {
                    "state": {
                      "domain": "adobedemo.com",
                      "cookiesEnabled": true
                    }
                  }
                }
              }
            }
          ]
        }
      }
    }
  ]
}
```

10. Click the code in the Request Payload section to highlight the complete code in blue, as shown:

The screenshot shows the Network tab with several requests listed. The 'Request Payload' section for one request is expanded, showing a large JSON object representing the event payload. The entire JSON content is highlighted with a blue background.

```
{
  "events": [
    {
      "xdm": {
        "eventType": "web.webpagedetails.pageViews",
        "web": {
          "events": [
            {
              "xdm": {
                "eventType": "web.webpagedetails.pageViews",
                "web": {
                  "meta": {
                    "state": {
                      "domain": "adobedemo.com",
                      "cookiesEnabled": true
                    }
                  }
                }
              }
            }
          ]
        }
      }
    }
  ]
}
```

11. Select the code, as shown, by double-clicking the first character and pressing Shift + down arrow keys until the complete code is selected. You can also triple-click the code to select the entire code at once.



The screenshot shows the Network tab of a browser developer tools interface. A specific JSON payload is selected, highlighted with a red box. The payload contains event details, web page details, device details, environment details, and implementation details. The selected code is the 'implementationDetails' object.

```

{
  "event": {
    "type": "web.webpageloaded.pageview",
    "webPageDetails": {
      "url": "https://builder.adobedemo.com/run/kaelhr/vt/home",
      "name": "home",
      "viewName": "home",
      "pageViews": [
        {
          "value": 1
        }
      ],
      "webReferrer": {
        "url": "https://auth.services.adobe.com/"
      },
      "device": {
        "screenHeight": 1080,
        "screenWidth": 1920,
        "screenOrientation": "landscape",
        "environment": {
          "type": "browser",
          "browserDetails": {
            "viewPortWidth": 768,
            "viewPortHeight": 750
          },
          "placeContext": {
            "longitude": -2022.89,
            "latitude": 19118.44,
            "localTimezoneOffset": -330
          },
          "timestamp": "2019-09-19T05:14:27.482Z"
        },
        "implementationDetails": {
          "name": "https://ns.adobe.com/experience/reactor",
          "version": "2.12.0(2.13.0)",
          "environment": {
            "browser": {
              "adobePlatformMap": {
                "identification": {
                  "core": {
                    "id": "1795104872264620589433686992494529458"
                  },
                  "demoEnvironment": {
                    "brandName": "kaelhr_jay"
                  }
                },
                "implementationDetails": {
                  "core": {
                    "channel": "web"
                  }
                }
              },
              "query": {
                "personalization": {
                  "schemas": [
                    "https://ns.adobe.com/personalization/default_content_item",
                    "https://ns.adobe.com/personalization/html_content_item",
                    "https://ns.adobe.com/personalization/json_content_item",
                    "https://ns.adobe.com/personalization/redirect_item",
                    "https://ns.adobe.com/personalization/doc_action"
                  ],
                  "decisions": [
                    "visua"
                  ]
                }
              }
            },
            "identity": {
              "fetch": [
                "F010"
              ]
            }
          }
        }
      }
    }
  }
}

```

12. Right-click the code and click **Copy** or press the Ctrl+C keys to copy the code.
13. Open the **JSON formatter** webpage from the Lab URLs document and paste the copied code on the left pane.
14. Click **Format / Beautify**. A readable version of the code is displayed on the right pane. Scroll through the content.

## Task 2: Verify the data in Experience Platform

1. Navigate to <https://platform.adobe.com/>. The **Adobe Experience Platform** page opens.
2. On the left pane, navigate to **DATA MANAGEMENT > Datasets**. The **Datasets** page opens.
3. Click the **Browse** tab to view the available datasets.
4. In the **Search** box, type **website**. The search results are displayed.
5. Click **Demo System - Event Dataset for Website (Global v1.1)**. The selected dataset opens on a new page. The page displays the recently ingested data for this dataset, as shown:



6. At the upper right, click **Preview dataset**. A preview of the ingested data is displayed.
7. Click **Close** at the upper right to close the preview and return to the dataset.

## Exercise 2: Ingest data by using a CSV file mapped to the XDM schema

---

In this exercise, you will perform the following tasks:

1. Create a CRM dataset by using a data generator tool and download as a CSV file
2. Map the CSV file to an XDM schema by using a workflow

Task 1: Create a CRM dataset by using a data generator tool and download as a CSV file

1. Open the Mockaroo URL, <https://mockaroo.com/6ec17090> in a Google Chrome browser. The link is also available in the Lab URLs PDF file within the exercise folder provided to you. The **mockaroo** website opens with the following fields:

- > first\_name
- > last\_name
- > gender
- > email
- > birthdate
- > home\_latitude
- > home\_longitude
- > city
- > state\_region
- > country\_code
- > country
- > mobile
- > zip
- > crmID

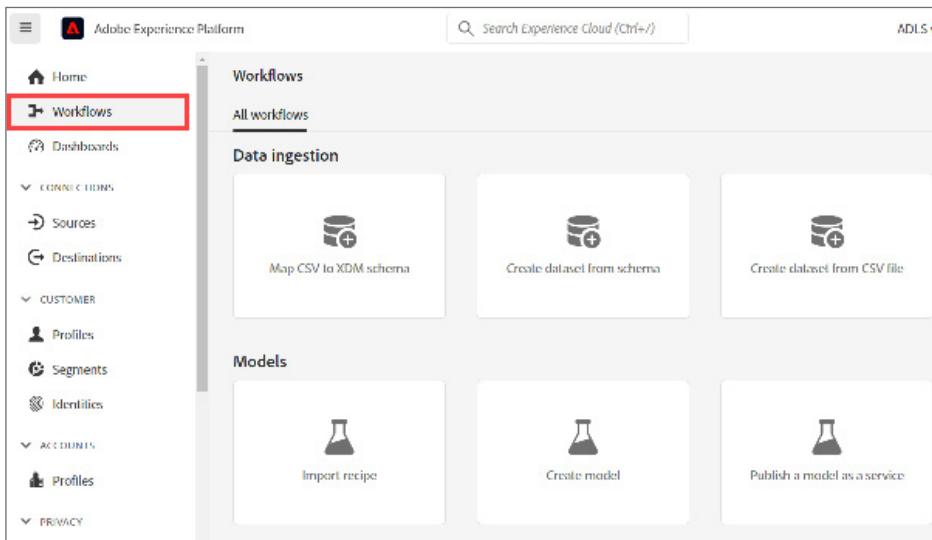
2. To generate a CSV file, click **DOWNLOAD DATA** at the bottom of the **mockaroo** website. A CSV file with rows of data is generated and saved to your computer.
3. Navigate to the location where the file was downloaded and rename the file by adding your initials before the extension. For example, **CRM\_CSV\_for\_DSN\_JD.csv**, where **JD** is your initials.



**Warning:** Do not open the CSV file in Microsoft Excel as it may change the format of the data.

## Task 2: Map the CSV file to an XDM schema by using a workflow

1. In Experience Platform, on the left pane, click **Workflows**, as shown. The **Workflows** page opens on the right with two sections, **Data ingestion** and **Models**.



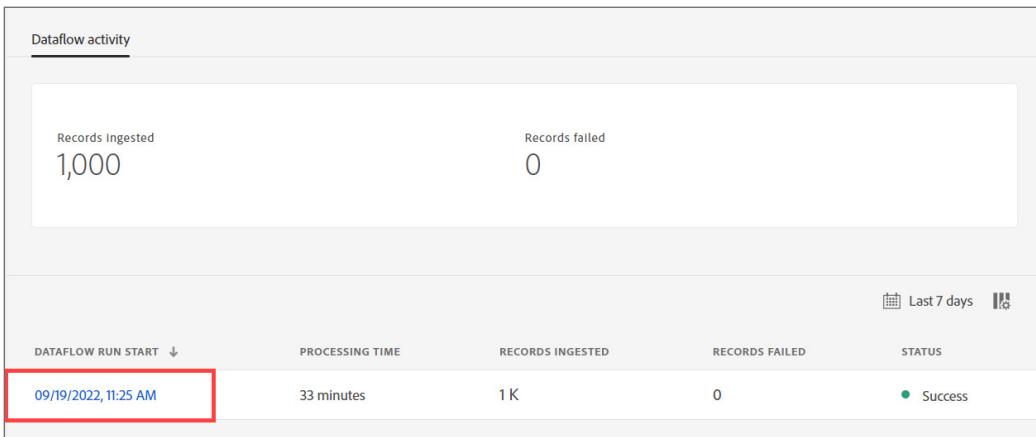
2. In the **Data ingestion** section, click **Map CSV to XDM schema**. The **Map CSV to XDM schema** pane opens on the right.
3. On the right pane, click **Launch** to start the process to upload the CSV file. The **Map CSV to XDM schema** page opens.

4. In the **Dataset details** section, ensure the **Existing dataset** option is selected, as shown:

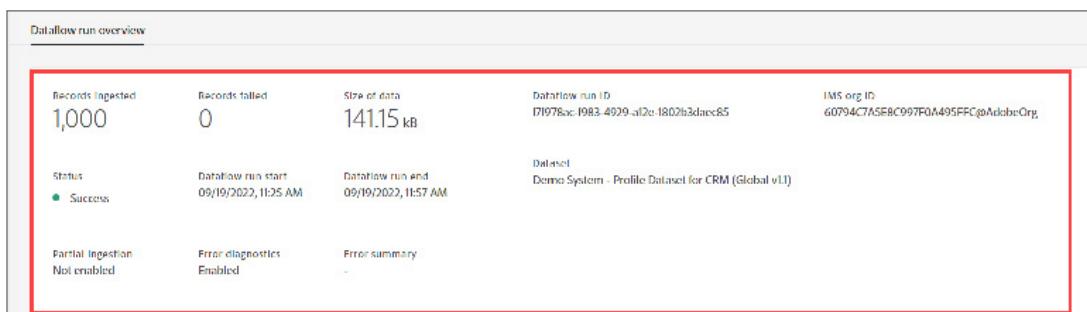
5. In the **Select dataset** box, type **CRM**. The search results are displayed.
6. Select the **Demo System - Profile Dataset for CRM (Global v1.1)** dataset. A green tick mark is displayed on the right of the dataset name.
7. In the **Dataflow details** section, in the **Dataflow name** box, enter <your initials>\_Today's date. For example, **JD\_0106**.
8. At the upper-right corner, click **Next**. The **Select data** section is displayed.
9. Click **Choose files** on the left to search for the CSV file on your computer.
10. Navigate to the location where you saved the file, select the file, and click **Open**. Alternatively, you can drag the file onto the **Drag and drop files** area. The file is uploaded, as shown:

|   | FIRST_NAME | LAST_NAME  | ECID                                   | GENDER | EMAIL                      |
|---|------------|------------|--|--------|----------------------------|
| 0 | Jessa      | Agates     | 39212167347806149888476981638564449199 | male   | jagates@ethay.co.uk        |
| 1 | Clemmy     | Daws       | 96702117949587002668696615765833735011 | male   | cdaws@thloglines.com       |
| 2 | Traver     | Addenbrake | 21586896244096859024579271081000286524 | female | taddenbrake2@marzmedia.com |

11. Click **Next**. The **Mapping** section opens. It may take some time for the file to process and open the **Mapping** section.  
In the **Mapping** section, you can map the source data to the destination schema fields and review the suggested mappings for accuracy. You can map each CSV column header (called the **Source Schema**) with a field in the dataset (called the target field).
12. Click **emailFormat** under **TARGET FIELDS**. A navigation tree opens on the right.
13. Scroll up to **adlsplatform<yourregion>**. For example, **adlsplatformapac**.
14. Expand **adlsplatform<your region> > identification > core** and select **email**. The **TARGET FIELDS** for email is updated with **\_adlsplatform<yourregion>.identification.core.email**.
15. Click **Finish** to complete the mapping. The **Sources > <Your Dataflow>** page opens. For example, **Sources > JD\_0106** page opens.
16. On the **Dataflow activity** tab, click the link, as shown. If you do not see the link, refresh the page a few times. Ensure the status is **Success**.



Notice the details in the **Dataflow run overview** tab, as shown:



17. Click the browser Back button to navigate back to the **Sources > <Your Dataflow>** page.
18. On the right pane, click the **Demo System - Profile Dataset for CRM (Global v1.1)** data link, as shown. The **Demo System - Profile Dataset for CRM (Global v1.1)** page opens.

The screenshot shows the 'Dataflow activity' page. On the left, there's a summary section with 'Records Ingested' (1,000) and 'Records failed' (0). Below this is a table with columns: DATAFLOW RUN START, PROCESSING TIME, RECORDS INGESTED, RECORDS FAILED, and STATUS. A single row shows '09/19/2022, 11:25 AM', '33 minutes', '1 K', '0', and 'Success'. To the right is a 'Properties' panel with fields for Dataflow name (PHP\_1909), Description, Source data, and Target dataset. The 'Target dataset' field contains 'Demo System - Profile Dataset for CRM (Global v1.1)', which is highlighted with a red box.

On this page, you can evaluate the status of the CSV file that is recently ingested into the dataset. You can scroll down to view the details of the data ingestion, such as batch ID, time, failures, and status.

19. Click **Preview dataset** at the upper-right corner of the page. The **Demo System - Profile Dataset for CRM (Global v1.1)** dialog opens with the preview data.
20. Click **adlsplatform<your region> > identification > emailId**. For example, **adlsplatformapac > identification > core > emailId**. The email IDs are displayed on the right.
21. Click **Close** to close the dialog box.

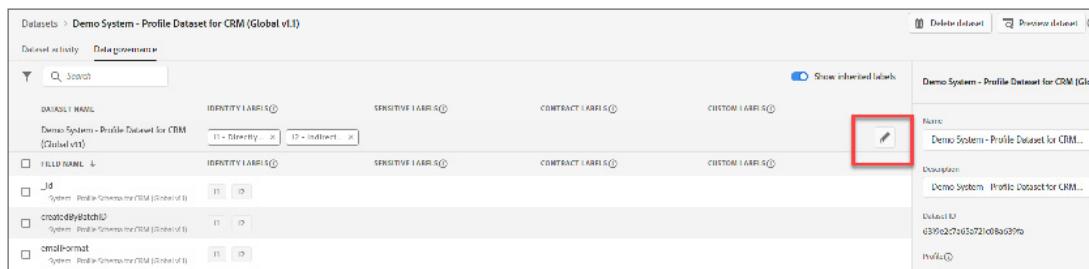
## Exercise 3: Apply data governance to a dataset

After the data is uploaded to Experience Platform, data governance can be applied to the dataset.

1. On the left pane, navigate to **DATA MANAGEMENT > Datasets**. The **Datasets** page opens.
2. Click the **Browse** tab, if not already open.
3. In the **Search** box, type **CRM**. The search results are displayed.
4. Select **Demo System - Profile Dataset for CRM (Global v1.1)**. The **Dataset activity** tab of the selected dataset opens.
5. Click the **Data governance** tab. The data governance applied to the dataset is displayed.

 **Note:** You can set three types of restrictions—Contract, Identity, and Sensitive. You can set governance labels at the dataset level, which will then be inherited by all fields. You can also specify governance labels at the field level. At the field level, you can select all field names, at once, by clicking the check box beside the **FIELD NAME** column header. You can also select one or more fields by clicking the check box beside each field name.

6. Click the pencil icon at the right of the **Demo System - Profile Dataset for CRM (Global v1.1)** column, as shown. The **Edit governance labels** dialog opens, with the **I1** and **I2** labels selected by default.



 **Note:** You need to zoom out to see the pencil icon.

7. If the labels are preselected, click **Cancel** to return to the **Data governance** tab.  
If the labels are not selected, select the **I1** and **I2** labels and click **Save changes** to return to the **Data governance** tab.
8. In the **IDENTITY LABELS** column, notice that each field has inherited the **I1** and **I2** labels, as shown:

| Dataset activity      Data governance  |   |                    |
|--|---|--------------------|
|  | IDENTITY LABELS ⓘ                         | SENSITIVE LABELS ⓘ |
| DATASET NAME   |   |                    |
| Demo System - Profile Dataset for CRM (Global v1.1)  | [ I1 - Directly... ] [ I2 - Indirect... ] |                    |
| <input type="checkbox"/> FIELD NAME ↴  | IDENTITY LABELS ⓘ                         | SENSITIVE LABELS ⓘ |
| <input type="checkbox"/> <b>Id</b><br>...m - Profile Schema for CRM (Global v1.1)                  | <b>I1</b> <b>I2</b>                       |                    |
| <input type="checkbox"/> <b>createdByBatchID</b><br>...m - Profile Schema for CRM (Global v1.1)    | <b>I1</b> <b>I2</b>                       |                    |
| <input type="checkbox"/> <b>emailFormat</b><br>...m - Profile Schema for CRM (Global v1.1)         | <b>I1</b> <b>I2</b>                       |                    |
| <input type="checkbox"/> <b>identityMap</b><br>...m - Profile Schema for CRM (Global v1.1)         | <b>I1</b> <b>I2</b>                       |                    |
| <input type="checkbox"/> <b>identityPrivacyInfo</b><br>...m - Profile Schema for CRM (Global v1.1) | <b>I1</b> <b>I2</b>                       |                    |

9. Click the **FIELD NAME** check box, as shown, to select all fields. The section on the right displays **I1** and **I2** identity labels and the number of fields that are selected.

| DATASET NAME  | IDENTITY LABELS ⓘ                         | SENSITIVE LABELS ⓘ |
|---|---|--------------------|
| Demo System - Profile Dataset for CRM (Global v1.1)   | [ I1 - Directly... ] [ I2 - Indirect... ] |                    |
| <input checked="" type="checkbox"/> FIELD NAME ↴  | IDENTITY LABELS ⓘ                         | SENSITIVE LABELS ⓘ |
| <input checked="" type="checkbox"/> <b>Id</b><br>...m - Profile Schema for CRM (Global v1.1)                  | <b>I1</b> <b>I2</b>                       |                    |
| <input checked="" type="checkbox"/> <b>createdByBatchID</b><br>...m - Profile Schema for CRM (Global v1.1)    | <b>I1</b> <b>I2</b>                       |                    |
| <input checked="" type="checkbox"/> <b>emailFormat</b><br>...m - Profile Schema for CRM (Global v1.1)         | <b>I1</b> <b>I2</b>                       |                    |
| <input checked="" type="checkbox"/> <b>identityMap</b><br>...m - Profile Schema for CRM (Global v1.1)         | <b>I1</b> <b>I2</b>                       |                    |
| <input checked="" type="checkbox"/> <b>identityPrivacyInfo</b><br>...m - Profile Schema for CRM (Global v1.1) | <b>I1</b> <b>I2</b>                       |                    |

10. Clear the checkbox beside the **FIELD NAME** column header.

11. Click one or more check boxes under **FIELD NAME** to select the individual fields. The section on the right displays **I1** and **I2** identity labels and the number of fields that are selected. The labels are inherited because they were specified at the dataset level. You can apply additional labels to the selected fields in the dialog box.
12. Click the **i** icon next to **SENSITIVE LABELS** and **CONTRACT LABELS** titles and read the descriptions.
13. Click anywhere on the page to close the descriptions.

## Module 3

# Managing Real-Time Customer Profiles in Adobe Experience Platform

## Introduction

Customers interact with brands through various channels that results in creation of different identities. It is challenging to combine these identities and create a single view of a customer.

Adobe Experience Platform enables you to bring together known and unknown customer data to create a consolidate view of a customer. You can create accurate customer segments and provide the right experiences to the customer.

## Objectives

After completing this course, you will be able to:

- Explain Real-time Customer Profile in Experience Platform
- Create a customer profile in Experience Platform
- Authenticate Postman to Adobe I/O and execute application programming interface (API) calls
- Verify profiles in Experience Platform
- Add cross-device data to Experience Platform
- Create a segment in Experience Platform by using the Experience Platform user interface (UI)
- Create a segment in Experience Platform by using API

# Real-Time Customer Profile: An Overview

---

Real-time Customer Profile enables you to:

- Provide consistent and relevant experiences to your customers irrespective of the channels they use to interact with your brand
- Create a holistic view of each individual customer by combining data from multiple channels including online, offline, customer relationship management (CRM), and third-party data
- Consolidate your siloed customer data with an actionable and timestamped account of every customer interaction

## Identity Service

Real-time Customer Profile uses Identity Service to:

- Stitch the fragmented customer identities
- Build a complete picture of a customer and their interactions with brands
- Link identities from multiple channels and create an identity graph for each customer

The two types of identities are:

- Known identity: An identity such as a login ID, Experience Cloud ID (ECID), or loyalty ID that can directly identify a person.
- Unknown or anonymous identities: An identity such as a visitor's IP address and cookie ID that single out a device without identifying the actual person using it.

The fragmented customer identities are reconciled into a single entity with the help of:

- Identity namespace: An identifier, such as a device ID or an email ID, used to provide the context from which a customer data originates.
- Identity graphs: A map of relationships between different identity namespaces. It provides a visual representation of customer interaction across channels.

## Segmentation Service

Experience Platform enables you to create segments and generate audiences from your Real-time Customer Profile data. You can create segments by using the Experience Platform UI or Representational State Transfer (RESTful) application programming interface (API).

The segments are centrally configured and maintained in Experience Platform and are accessible to any Adobe solution.

# Exercise 1: Create a customer profile in Experience Platform

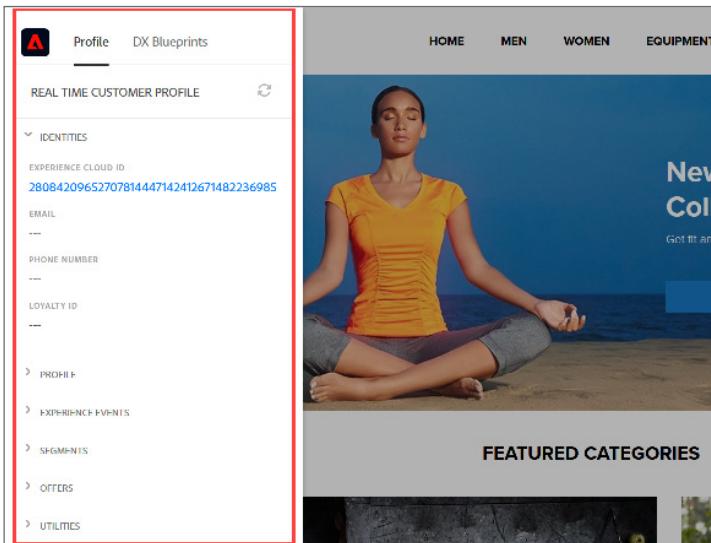
To create a profile in Experience Platform with different ECIDs, you will perform the following tasks:

1. Interact with a website anonymously
2. Register on the website and create a profile

## Task 1: Interact with a website anonymously

In this task, you will interact with the **LUMA** website anonymously and create an identity.

1. Open the Lab URL document for your region and copy the **LUMA** website link.
2. Open an incognito window in your browser, paste the website link and press the Enter key. The **LUMA** website opens.
3. At the upper-left corner, click the Adobe logo. A pane opens, as shown:



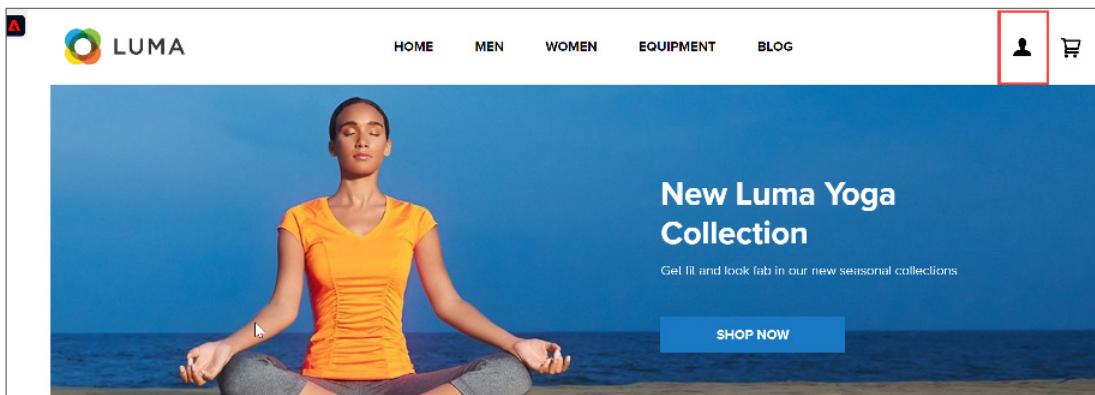
Notice that an Experience Cloud ID is available in the **IDENTITIES** section.

4. Copy the **EXPERIENCE CLOUD ID** value and paste it in a text document on your computer. You will use this ID in Exercise 2.
5. Expand the **PROFILE** section and notice that the fields are currently blank.
6. Click anywhere on the website to close the left pane.

## Task 2: Register on the website and create a profile

In this task, you will register your profile in the LUMA website through an incognito window of the browser.

1. In the LUMA website, click the account icon at the top, as shown. The customer login and registration page opens.



2. Click the three vertical dots on the upper-right corner of the browser and navigate to **More tools > Developer tools**. The developer tool dialog opens.
3. On the website, click **CREATE AN ACCOUNT**. The **CREATE AN ACCOUNT** page opens.
4. Update the following details. Do not include your personal information. Use fake details.
  - a. In the **First Name** box, type a first name. For example, **Student**.
  - b. In the **Last Name** box, type a last name. For example, **One**.
  - c. Click the **Shoe size** drop-down menu and select a shoe size. For example, **44**.
  - d. In the **Email** box, type an email ID. For example, **studentone@training.com**.
  - e. In the **Mobile number** box, type a mobile number. For example, **1234567890**.
  - f. Click the **Shirt size** drop-down menu and select a shirt size. For example, **L**.
  - g. Click the **Gender** drop-down menu and select a gender. For example, **Female**.
  - h. In the **Birth Date** box, type a date of birth, or select it from the calendar. For example, **09/01/2000**.
  - i. Click the **Preferred color** drop-down menu and select any color. For example, **Yellow**.
  - j. In the **Street Address and Number** box, type an address. For example, **St. 1**.
  - k. In the **City** box, type any city name. For example, **Tokyo**.
  - l. In the **Postal Code** box, type the postal code. For example, **56001**.
5. Ensure all the notification and policy check boxes are selected and click **Register**. The **Thank you for registering** message is displayed at the top.

- To view the distributed control system (DCS) call from the **LUMA** website to Experience Platform, ensure you are in the **Network** tab and type **dc** in the **Filter** box. The DCS call is displayed, as shown:

The screenshot shows a LUMA web application interface. On the left, there's a blue sidebar with the title 'JOIN US' and a message: 'Gain access to the latest news and limited benefit-based live documents.' Below this is a 'CREATE AN ACCOUNT' button. The main content area has a 'WELCOME BACK' heading and three input fields: 'Email or Username', 'Password', and 'Remember me'. A large blue 'SIGN IN' button is centered at the bottom of the form. To the right, a browser's developer tools are open, specifically the Network tab. It lists several requests, including one to 'https://luma-test.com/api/auth/login' with a status of 'Success' and a response size of 20000 ms. Other requests listed include 'https://luma-test.com/api/auth/logout', 'https://luma-test.com/api/auth/me', and 'https://luma-test.com/api/auth/me'. The developer tools also show the 'Request Payload' for the login request, which includes a JSON object with fields like 'username', 'password', and 'remember\_me'.

A small icon of a pencil inside a square frame, indicating a text input field.

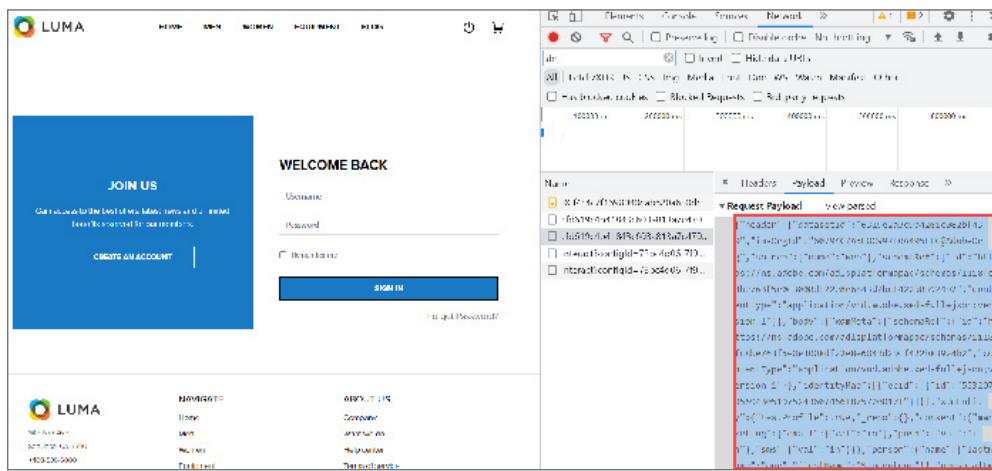
**Note:** The number of calls may differ based on the activities that you perform on the website.

7. Click a DCS call. The call details are displayed on the right.

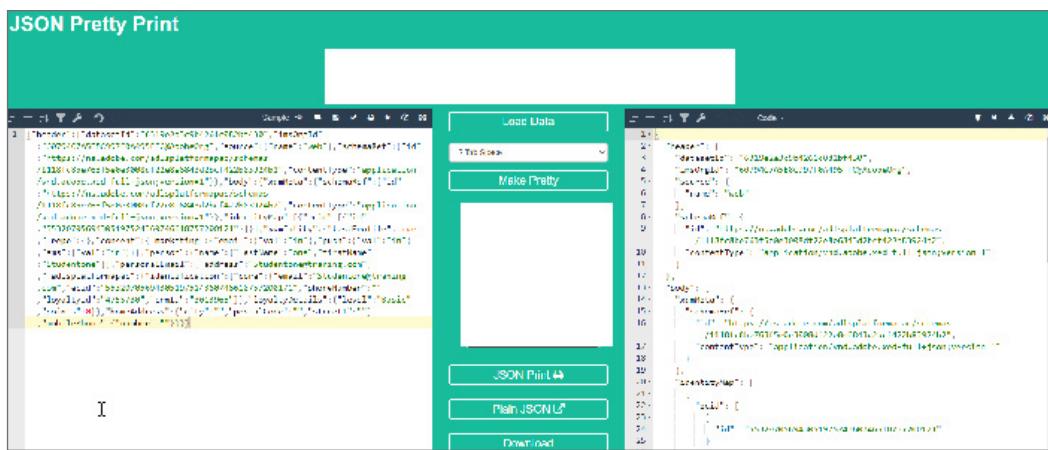
8. In the **Payload** tab, click **view source**, as shown. The code is displayed.

The screenshot shows a web application interface. At the top, there's a navigation bar with links for HOME, MEN, WOMEN, EQUIPMENT, and BLOG. To the right of the navigation are icons for power, refresh, and search. Below the navigation is a large blue header section with the text 'JOIN US' and a subtext 'Gain access to the best offers, latest news and unlimited benefits reserved for our members.' A 'CREATE AN ACCOUNT' button is located at the bottom left of this section. The main content area has a title 'WELCOME BACK' and two input fields for 'Username' and 'Password'. There's also a 'Remember me' checkbox and a 'SIGN IN' button. To the right of the main content, the browser's developer tools are open, specifically the Network tab. This tab lists several requests, with one particular request highlighted: 'joinconfig?configId=73bcb006-795f-4...'. The status of this request is 'Pending'. Other tabs in the developer tools include Elements, Console, Sources, and Network.

9. Click the code. The complete code highlights in blue.
10. Press Ctrl+A to select the entire code. The selected code is highlighted in light blue color, as shown:



11. Right-click the selected code and click **Copy** or press the **Ctrl+C** keys to copy the code.
12. Open <https://jsonformatter.org/> in a browser. The JSON formatter webpage opens.
13. Paste the copied code on the left. The output is displayed on the right, as shown:



**Note:** If the box on the left is prepopulated, delete the existing code and paste the new code.



**Note:** If the output is not displayed, proceed to the next step.

14. (Optional) Click **Make Pretty** to arrange the code in a readable format on the right. The call content is updated on the right.

15. Scroll down to view the identification details such as ecid and email, as shown:

```
_adlsplatformapac": {  
    "identification": {  
        "core": {  
            "email": "Studentone@traning.com",  
            "ecid": "55320705694305197524360746618757200171"
```

You can also find these values on the left pane of the **LUMA** website.

16. To check the values in the **LUMA** website:

- a. Click the Adobe logo at the top left to expand the pane.
- b. View the details under **IDENTITIES** and **PROFILE** sections.

## (Optional) Exercise 2: Authenticate Postman to Adobe I/O to execute API calls

This exercise includes the following tasks:

1. Configure Postman for API calls to Experience Platform
2. Retrieve the profile data from Experience Platform by using an API call
3. Retrieve the Experience Data Model (XDM) ExperienceEvent data from Experience Platform by using an API call

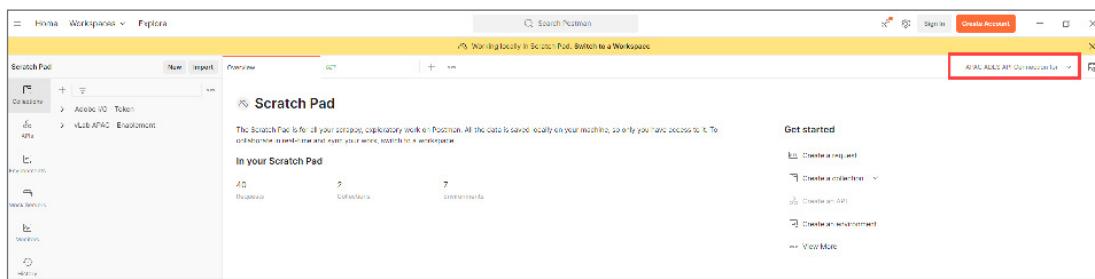
### Task 1: Configure Postman for API calls to Experience Platform

1. Click the Postman icon on the task bar. The **Postman** application opens.
2. At the upper-left, click **Import**. The **Import** dialog opens.
3. Click **Upload Files** and navigate to the location where the lab files specific to your region are saved.
4. Open the Postman APIs folder, press Shift and select the three json files, as shown, and click **Open**. The **Import** dialog opens.

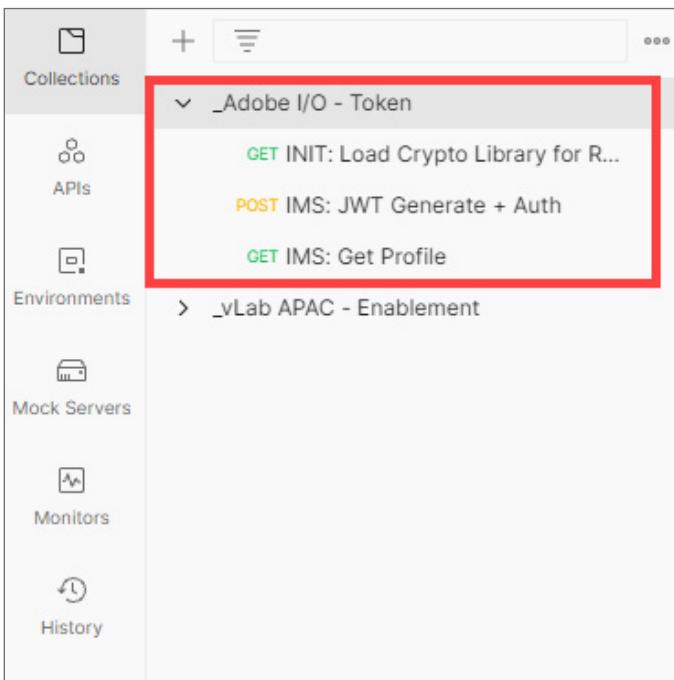


5. Click **Import**. The files are uploaded to Postman.

- At the right, click the **No environment** drop-down arrow and select the environment specific to your region, as shown. For example, **\_APAC ADLS API Connection**.



- Click the **Collections** tab on the left.
- On the left pane, click the **\_Adobe I/O - Token** API collection, as shown. The available calls are displayed.



You need to call an external library that takes care of the encryption and decryption of the communication with Experience Platform. To load this library, you must execute the **INIT: Load Crypto Library for RS256** call.

9. Click **INIT: Load Crypto Library for RS256**. The **GET INIT: Load Crypto Library for RS256** tab opens on the right.

10. Click **Send**. The result of the call is displayed on the **Body** tab, as shown:

```

{
  "key": "JwksSignature",
  "value": "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAQIDQABRzXWZLcOvDfCgqBz\n-----END PUBLIC KEY-----"
}
  
```

With the crypto library now loaded, you can authenticate Postman to Adobe I/O.

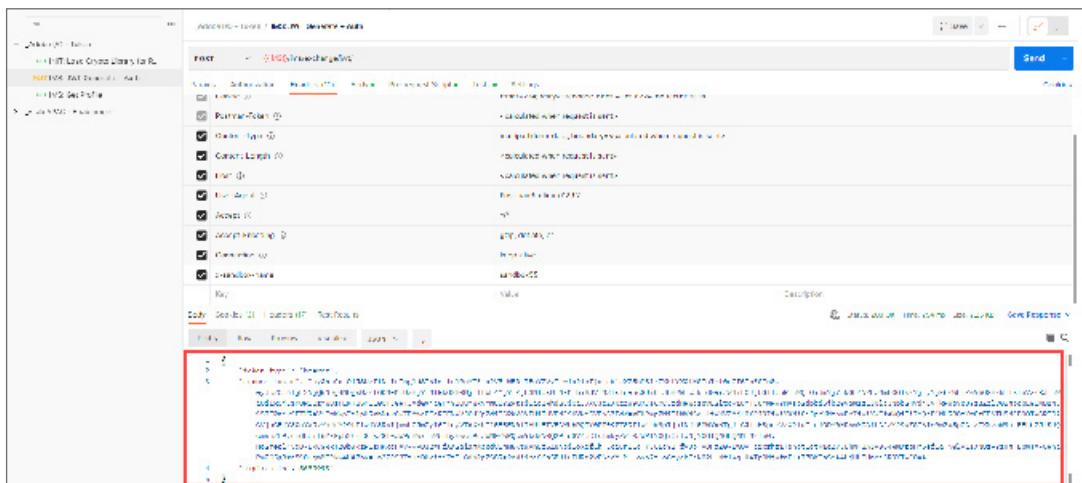
11. On the left pane, click the **IMS: JWT Generate + Auth** call. The **POST IMS: JWT Generate + Auth** tab opens on the right.

12. Click the **Headers** tab and add the following details, as shown:

- In the **Key** column, type **x-sandbox-name**.
- In the **Value** column, type the sandbox number provided to you by your instructor. For example, **sandbox55**.

| Key            | Description |
|----------------|-------------|
| x-sandbox-name | sandbox55   |

13. Click **Send**. The result of the call is displayed, as shown:



If your configuration is successful, you will see a response that contains **token\_type** and **access\_token**. Adobe I/O has provided a bearer-token, with a specific value and an expiration window in milliseconds. The token that you have received is valid for 24 hours. This means that after 24 hours, if you want to use Postman to authenticate to Adobe I/O, you will have to generate a new token by running this call again. Copy the token and paste it in a notepad for easy access.

## Task 2: Retrieve the profile data from Experience Platform by using an API call

1. On the left pane, click the **IMS: Get Profile** call. The **GET IMS: Get Profile** tab opens on the right.
2. Click the **Headers** tab and add the following details:
  - a. In the **Key** column, type **x-sandbox-name**.
  - b. In the **Value** column, type the sandbox number provided to you by your instructor. For example, **sandbox55**.
3. Click **Send**. The result of the call is displayed on the **Body** tab, as shown. If the result is generated without any error, you are connected to the correct Experience Platform instance.

The screenshot shows the Adobe Experience Platform API browser interface. The left sidebar lists various API endpoints under 'IMS: Get Profile'. The main area shows a successful response for the 'GET' method. The 'Headers' tab shows a single header 'Content-Type: application/json'. The 'Body' tab displays a JSON object representing a customer profile:

```

{
  "id": "2020-04-22T12:42:42Z-1000000000000000000",
  "version": "2020-04-22T12:42:42Z-1000000000000000000",
  "name": "John Doe",
  "lastName": "Doe",
  "middleName": null,
  "firstName": "John",
  "suffix": null,
  "gender": "M",
  "dateOfBirth": "1990-01-01T00:00:00Z",
  "email": "john.doe@example.com",
  "phone": "+1234567890",
  "addresses": [
    {
      "id": "2020-04-22T12:42:42Z-1000000000000000001",
      "line1": "123 Main Street",
      "line2": "Apt 456",
      "city": "Anytown",
      "state": "CA",
      "zip": "90210"
    }
  ],
  "tags": [
    "Customer"
  ],
  "lastModified": "2020-04-22T12:42:42Z-1000000000000000000"
}
  
```

4. On the left pane, click **\_vLab <yourregion> - Enablement** to expand the collection. For example, **\_vLab APAC - Enablement**. Two subcollections are displayed.
5. Click **1. Unified Profile Service**. The available calls in the collection are displayed.
6. Click the **UPS - GET Profile by Entity ID & NS** call. The **GET UPS - GET Profile by Entity ID & NS** tab opens on the right.

The three required variables for the call are mentioned in the table below:

| Key         | Value                | Description   |
|-------------|----------------------|---|
| entityID    | id                   | The specific customer ID                                      |
| entityIdNS  | namespace            | The specific namespace that is applicable to the ID           |
| schema.name | _xdm.context.profile | The specific schema for which you want to receive information |

7. In the **VALUE** column of the **entityID** key, select the existing value and delete it.

8. Type the ECID value that you copied in Task 1 of Exercise 1 and paste it in the **VALUE** column of the **entityID** key, as shown:

The screenshot shows a Postman collection named "Adaptive (V2) - Tokens". A specific request is selected: "UPS - Get Profile by Entity ID & NS". The method is set to "GET" and the URL is `https://platform.module.idvcloudaccess.com/api/v1/entityId=553207050430510752436074661875720017/entityNS=oid&schemaName=vdm.context.profile`. The "Authorization" tab is active, showing a Bearer token. The "Headers" tab includes "Content-Type: application/json" and "Accept: application/json". The "Body" tab is empty. The "Tests" and "Settings" tabs are also visible.

9. Click the **Headers** tab and add the following details:
    - a. In the **Key** column, type **x-sandbox-name**.
    - b. In the **Value** column, type the sandbox number provided to you by your instructor. For example, **sandbox55**.
  10. Click **Send** to send your request to Experience Platform. The result of the call is displayed on the **Body** tab, as shown:

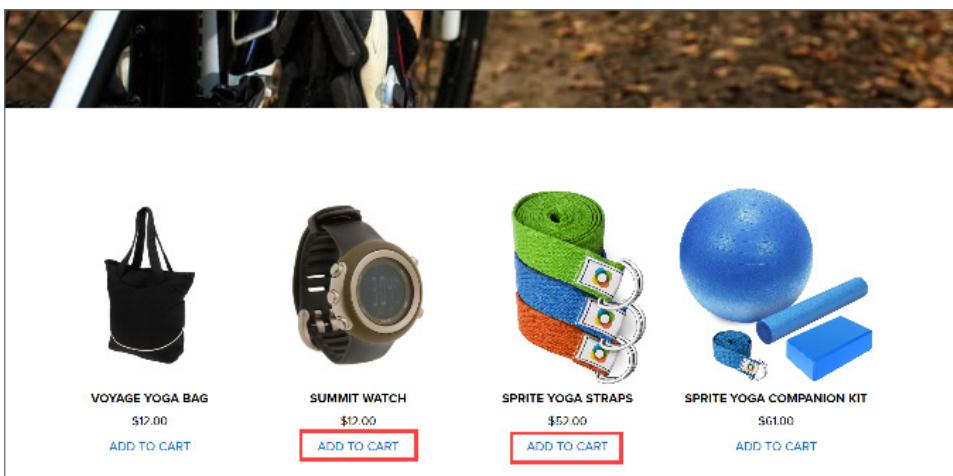
The result displays all available profile data in Experience Platform for **ecid**. You can use any ID such as mobile number or email ID to request this data.

11. To retrieve the profile information for a specific email ID by using the Experience Platform API, perform the following steps:
  - a. click the **Params** tab.
  - b. In the **Value** box for the **entityID** key, delete the existing value and type your email ID that you provided when registering your profile in the **LUMA** website.
  - c. In the **Value** box for the **entityIdNS** key, delete the existing value and type **email**.
  - d. In the **Value** box for the **schema.name** key, retain the existing value.
12. Click **Send**. The results are displayed on the **Body** tab. Scroll through the contents of the body tab and notice details such as email, loyaltyId, ecid, and crmid.

### Task 3: Retrieve the XDM ExperienceEvent data from Experience Platform by using an API call

You need to interact with the **LUMA** website to generate some event data for this task. To do so:

1. Log in to the **LUMA** website with your registered ID and password that you used to create the profile in Task 2 of Exercise 1. The **LUMA** website opens.
2. Scroll down and click one of the featured categories. For example, **Equipment**. The product page opens.
3. Add two products to the cart by clicking **ADD TO CART** under respective products, as shown. For example, **SUMMIT WATCH** and **SPRITE YOGA STRAPS**. The products are added to the cart.



To query Experience Platform's APIs for XDM ExperienceEvent data:

4. Open the **Postman** application.
5. On the left pane, expand the **\_vLab <yourregion> - Enablement** collection. For example, **\_vLab APAC - Enablement**.
6. Click **1. Unified Profile Service**. The available calls are displayed.
7. Click the **UPS - GET Profile & EE by Entity ID & NS** call. The **UPS - GET Profile & EE by Entity ID & NS** tab opens on the right.

The available parameters for the call are:

- › **schema.name**: Is the specific schema for which you want to receive information.
- › **relatedSchema.name**: Is the identity for which the data must be received.
- › **relatedEntityId**: Is the specific customer ID.
- › **relatedEntityIdNS**: Is the specific namespace that is applicable to the ID.

8. To retrieve profile information for your ECID by using the Experience Platform API, perform the following steps, as shown:
- In the **Value** box for the **schema.name** key, retain the existing value.
  - In the **Value** box for the **relatedSchema.name** key, retain the existing value.
  - In the **Value** box for the **relatedEntityId** key, delete the existing value and type the ECID that you copied in Task 1 of Exercise 1. You can also copy the ECID from the LUMA website.
  - In the **Value** box for the **relatedEntityIdNS** key, retain the existing value.

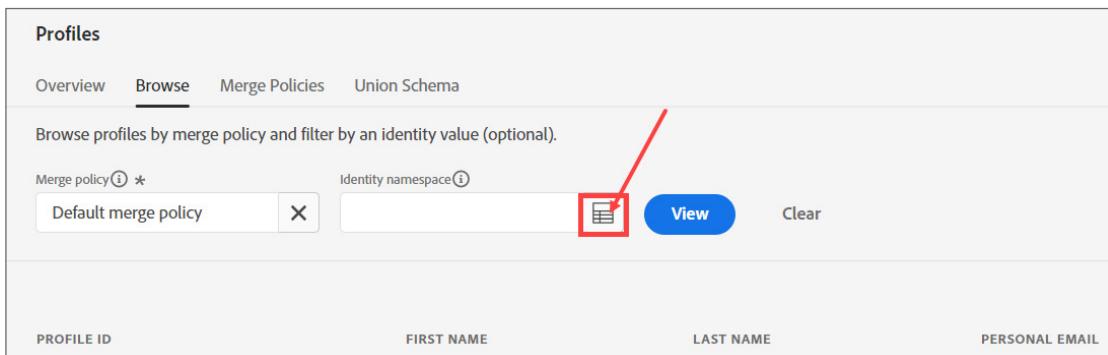
| KEY  | VALUE                                  | DESCRIPTION |
|--|--|-------------|
| <input checked="" type="checkbox"/> schema.name        | xrm.lead.customer                      |             |
| <input checked="" type="checkbox"/> relatedSchema.name | xrm.customer.profile                   |             |
| <input checked="" type="checkbox"/> relatedEntityId    | 593207050048031075248007466/8737200171 |             |
| <input checked="" type="checkbox"/> relatedEntityIdNS  | xid                                    |             |
| Key  | Value                                  | Description |

9. Click the **Headers** tab and add the following details:
- In the **Key** column, type **x-sandbox-name**.
  - In the **Value** column, type the sandbox number provided to you by your instructor. For example, **sandbox55**.
10. Click **Send** to execute the call. The result is displayed on the **Body** tab.

## Exercise 3: Verify profiles in Experience Platform

In this exercise, you will search a profile in Experience Platform by using an identity namespace and a corresponding value for the namespace.

1. Log in to Experience Platform. The **Home** page opens.
2. On the left pane, click **CUSTOMER > Profiles**. The **Profiles** page opens.
3. On the **Browse** tab, notice that the **Merge policy** box is updated with **Default merge policy** by default.
4. Click the list icon in the **Identity namespace** box, as shown. The **Select identity namespace** dialog opens.



5. In the **Search** box, type **Email**. The search results are displayed.
6. Select **Email** from the list and click **Select**. The **Identity namespace** box is updated with **Email** value.
7. In the **Identity value** box, type the email you used to register in the website.
8. Click **View**. A profile matching the identity value is listed.

9. Click the link in the **PROFILE ID** column, as shown. The profile details open.

| PROFILE ID                         | FIRST NAME | LAST NAME | PERSONAL EMAIL          |
|------------------------------------|------------|-----------|-------------------------|
| Bv7qrwVy8bLbnXp7aJ3vra2op4p4P3KjgA | Student    | One       | studentone@training.com |

10. Click the **Events** tab, as shown. The events associated with the ID is displayed. You can now view and examine the Real-time Customer Profile in Experience Platform.

| ACCOUNT | VALUE                             | PIVOT                                   |
|---------|-----------------------------------|---|
| channel | web                               | _apple_impression_on_device             |
| latency | 4min-4Y                           | _apple_impression_on_device_and_latency |
| cid     | 327388152015866181132817251267790 | _apple_impressions_identification_error |
| ip4     | 49.205.140.223                    | environment_ip4                         |
| type    | browser                           | environment_type                        |

## Exercise 4: Add cross-device data to Experience Platform

---

This exercise includes the following tasks:

1. Log in to the Luma website on your mobile phone
2. Verify the profile in Experience Platform

### Task 1: Log in to the Luma website on your mobile phone

1. Open a Google Chrome browser on your mobile phone.
2. Type the **LUMA** website link in the browser from the Labs URL document specific to your region. The Luma website opens.
3. Click the Adobe logo at the top-left of the screen. A pane opens. Alternatively, swipe the screen to the right.
4. Under **IDENTITIES > EXPERIENCE CLOUD ID**, you will find the ECID. Notice that this ECID is different from ECID of the **LUMA** website on your computer.
5. Copy the ECID and paste it in a notepad or an email that you can access later.
6. Swipe the pane to the left to close it.
7. Click the account icon on the top right. The registration and sign in page opens.
8. Scroll down to the **WELCOME BACK** section, and in the **Username** field, type the user name you created in a previous exercise. For example, **studentone@training.com**.
9. In the **Password** field, type **1234**.
10. Click **Sign IN**. The **Home** page opens.
11. Click a category at the top of the page. For example, **Equipment**. The **Equipment** page opens.
12. Scroll down and tap on **ADD TO CART** under one of the products. For example, **Voyage Yoga Bag**. The product is added to the cart. Ensure to add a product that you did not select in the previous exercises. The product is added to the cart.

## Task 2: Verify the profile in Experience Platform

- In the Experience Platform UI left pane, navigate to **CUSTOMER > Profiles**. The **Browse** tab opens on the right.
- In the **Merge policy** box, notice that the **Default merge policy** value is updated by default.
- In the **Identity namespace** box, search for, and select **Email**. The **Identity namespace** box is updated.
- In the **Identity value** box, type the email ID you used to log in to the Luma website on your mobile device. For example, **studentone@training.com**.
- Click **View**. A profile link matching to the namespace is listed. If you do not see the profile link, wait for a few minutes, refresh the page, and repeat steps 3 to 5.
- Click the link in the **PROFILE ID** column. The profile page opens with the combined profile data associated with the ID.
- In the **Detail** tab, under the **Linked identities** section, observe that all ECIDs from your mobile devices and desktop browsers are displayed on the page, as shown:

The screenshot shows the 'Detail' tab of a customer profile. The 'Customer profile' section displays a placeholder image and the name 'Student One'. The 'Basic attributes' section includes fields for Address (studentone@training.com), Gender (female), Birth Day And Month (blank), Number (1234567890), and Street1 (St. 1). The 'Linked identities' section contains a table with columns for Id and Ecid. The 'Ecids' column lists several values, with the first one highlighted by a red box: 327098843716285660184432147225472677902, 335449779455404161961763910197067912, 200812096527070791112211261102236985. The 'Channel preferences' section lists Direct mail, Phone, SMS, Email, Facebook news feed, and Twitter feed.

| Id        | Ecids   |
|-----------|---|
| Luid      | 327098843716285660184432147225472677902<br>335449779455404161961763910197067912<br>200812096527070791112211261102236985 |
| Email     | studentone@training.com   |
| LoyaltyId | 6790011   |

- Click the **Events** tab to view the events from your mobile phone and computer devices.
- Click **View all** under **commerce.productListAdds**. The **Events** dialog box opens.

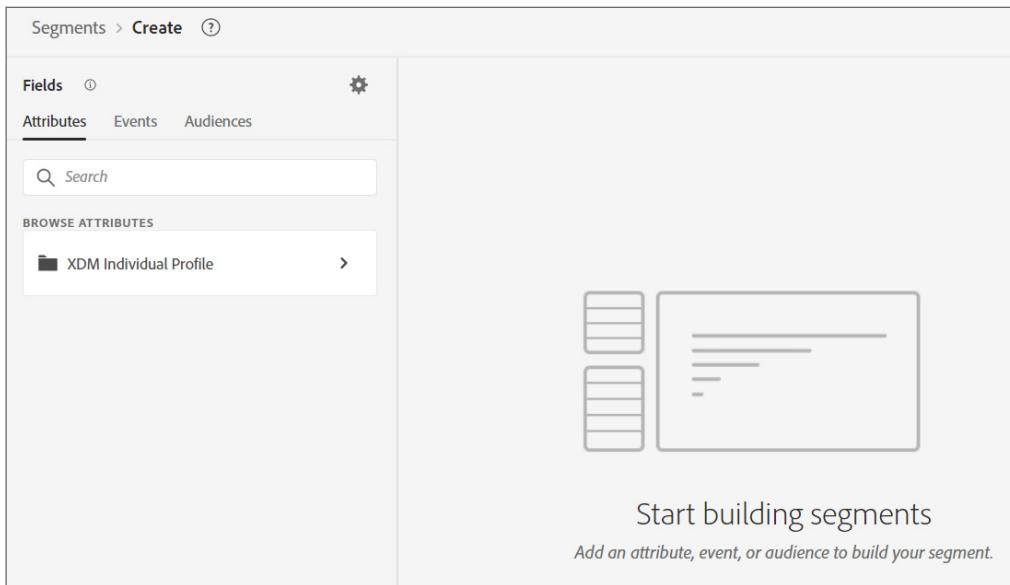
10. Scroll down the dialog box and notice the ecid, name, and mainCategory attributes as shown. This should match the mobile site ecid and the product you added to the cart on your mobile phone.

| ATTRIBUTE        | VALUE   | PATH   |
|------------------|---|--|
| mainCategory     | equipment   | productListItems.0._adisplatformapac.core.mainCategory |
| productURL       | https://builder.adobedemo.com/run/karthr-JZYT/product/  | productListItems.0._adisplatformapac.core.productURL   |
| ImageURL         | https://contentviewer.s3.amazonaws.com/helium/luma-y... | productListItems.0._adisplatformapac.core.ImageURL     |
| quantity         | 1   | productListItems.0.quantity                            |
| productAddMethod | web   | productListItems.0.productAddMethod                    |
| priceTotal       | 50  | productListItems.0.priceTotal                          |
| name             | Sprite Yoga Straps                                      | productListItems.0.name                                |
| SKU              | product/  | productListItems.0.SKU                                 |
| currencyCode     | USD   | productListItems.0.currencyCode                        |

## Exercise 5: Create a segment in Experience Platform by using the Experience Platform UI

In this exercise, you will create a segment to capture the Female profiles whose product add method is through mobile.

1. In the Experience Platform left pane, navigate to **CUSTOMER > Segments**. The **Segments** page opens.
2. Click the **Browse** tab. The existing datasets are displayed.
3. Click **Create segment**. The segment builder opens, as shown:



All data that is used to create segments in Experience Platform must be mapped against Experience Data Model (XDM), so that the data becomes part of the same data model regardless of the source of the data. Segments can be sent to solutions such as Adobe Target, Adobe Campaign, Adobe Audience Manager, or solutions outside Adobe for activation.

4. Expand **XDM Individual Profile**. The XDM attribute folders open.
5. Scroll down and expand the **Person** attribute. The available attributes are displayed, as shown:

The screenshot shows the 'Fields' interface with the 'Attributes' tab selected. The breadcrumb navigation shows 'XDM Individual Profile > Person'. A red box highlights the list of attributes under 'Person': 'Full name', 'Birth date (MM-DD)', 'Birth date(YYYY-MM-DD)', 'Birth year', and 'Gender'. Each attribute entry includes a folder icon, the attribute name, a description, and an information icon.

6. Drag **Gender** onto the **Start building segments** area. The **Attributes** section is displayed, as shown:

The screenshot shows the 'Fields' interface with the 'Attributes' tab selected. The breadcrumb navigation shows 'XDM Individual Profile > Person'. The 'Attributes' section is open, showing a condition: 'Gender equals Select or type options'. Below this, there is a note: 'At least one option required'. The 'Events' section is also visible below the attributes.

- In the **Attributes** section, click the **Select or type options** drop-down arrow and select a gender. For example, **Female**. The selected gender is displayed beside the **Select or type options** box, as shown:

The screenshot shows the 'Fields' interface with the 'Attributes' tab selected. In the 'Include' section, there is a rule for 'Gender': 'Gender equals Female'. The 'Female' option in the dropdown is highlighted with a red box.

- In the **Fields** section, click the **Events** tab. The page is updated to add the event data, as shown. The classes associated with the event are listed on this page.

The screenshot shows the 'Fields' interface with the 'Events' tab selected. The 'Events' section displays a timeline with a placeholder for 'Any time'. Below the timeline, there is a sequence of four dashed boxes connected by arrows, representing a workflow or event flow.

- In the **BROWSE CLASSES** section, expand **XDM ExperienceEvent**. The events are displayed.
- Scroll down and click **Product list items** to view the available attributes.

11. Drag **Product add method** onto the **Add events to build a timeline** area. The page is updated, as shown:

The screenshot shows the 'Event Rules' configuration screen. On the left, there's a sidebar with various XDM objects like 'Currency code', 'Line item ID', 'Name', 'Price total', 'Quantity', and 'SKU'. The main panel shows a timeline with a single event node labeled 'Any 1 Rule'. Below it, the 'Event Rules' section is expanded, showing a condition: 'Include At least 1 Any where:'. Underneath, another condition is shown: 'Include Any Product list items where:'. This condition includes a specific rule: 'Product add method equals Select or type options'. The dropdown menu 'Select or type options' is open, and the value 'Mobile' is selected, highlighted with a red box.

12. Retain the **equals** logical condition.

13. From the **Select or type options** drop-down menu, select **Mobile**. The selected value is displayed on the right, as shown:

This screenshot is a close-up of the 'Event Rules' configuration. It shows the 'Select or type options' dropdown menu open, with the value 'Mobile' selected and highlighted with a red box. The 'Case sensitive' checkbox is also visible below the dropdown.

14. On the right pane, in the **Name** field, type a name for the segment. For example, **StudentOne\_ProductAddMethod\_Segment**.

15. Click **Save** to create the segment. The **Segment is saved** message is displayed and the newly created segment details are displayed. This segment can be used later for data analysis and reporting.

## (Optional) Exercise 6: Create a segment in Experience Platform by using API

In this exercise, you will use API to create segments in Experience Platform.

This exercise includes the following tasks:

1. Create a new segment definition
2. Post a new segment job request
3. Get the segment job request status

### Task 1: Create a new segment definition

In this task, you will create a segment in Experience Platform by using API

1. Open the Postman application. Notice that the environment specific to your region is selected at the upper right. For example, **\_APAC ADLS API Connection**.
2. On the left pane, click **\_vLab <your region> - Enablement**. For example, **\_vLab APAC - Enablement**. The **Unified Profile Service** and **Segmentation** folders are displayed.
3. Expand **Segmentation**. The available calls in the folder are displayed.
4. Click **Step 1 - Profile: Create A Segment Definition**. The **POST Step 1 - Profile: Create A Segment Definition** tab opens on the right.
5. Click the **Headers** tab and add the following details:
  - a. In the **Key** column, type **x-sandbox-name**.
  - b. In the **Value** column, type the sandbox number provided to you by your instructor. For example, **sandbox55**.
6. Click the **Body** tab. The JSON/PQL code for the call is displayed.

7. In the **name** line item, replace the word **ldap** with your student ID. For example, **vlabaepapac2**, as shown. You can access the student ID from your Experience Platform profile.

The screenshot shows a software interface for managing API flows or workflows. At the top, there's a navigation bar with tabs like 'New', 'Import', 'Overview', and several 'SCT' related items. Below the navigation is a search bar and a tree view of workflow steps.

**Workflow Tree:**

- Acobe ID Token
  - INIT: Load Crypto Library for R...
  - POST MS: JWT Generate + Auth
  - POST MS: Get Profile
- vLab APAC - Enablement
  - POST UPS: GET Profile by Entity ID...
  - POST UPS: GET Profile & BE by Ent...
- vLab APAC - Segmentation
  - Step 1: Profile: Create A Seg...
  - Step 2: POST Segment Job
  - Step 3: GET Segment Job stat...

**Current Step Details:**

**Method:** POST    **URL:** {{APIURL}}CRM\_GATEWAY/texts/segment/definition

**Body:**

```
1. "name": "polyักษณ์", "All": "true", "CustomName": "(All)",  
2. "expression": { "type": "QL", "format": "poly/json", "value": "[{"method": "GET", "uri": "/v1/identity/individualLookup", "individual": "V", "password": "V", "object": "V", "methodType": "V"}, {"literal": "V", "value": "[{"formId": "V"}]"}], "knowledge": { "id": "V", "values": [ { "name": "V", "value": "V", "context": "V" } ] }, "tilinewya": "V"
```

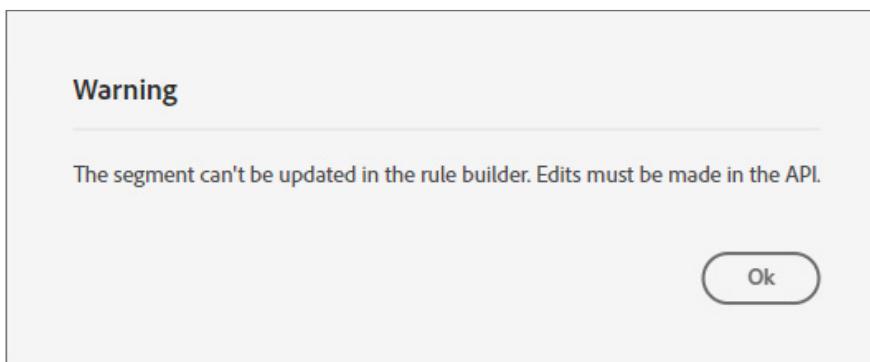
8. Click **Send** to create the segment.
  9. To verify if the segment was created in the Experience Platform UI:
    - a. In the Experience Platform UI, navigate to **CUSTOMER > Segments**. The **Browse** tab of the **Segments** page opens. The segment that you created is displayed, as shown:

| Segments             |           |       |               |                   |
|----------------------|-----------|-------|---------------|-------------------|
| NAME                 | BREAKDOWN | CHURN | PROFILE COUNT | EVALUATION METHOD |
| All Female Custom... | -         | -     | -             | Streaming         |

- b. Click the segment name that you created. The selected segment opens on a new page, as shown:

| Detail | Segment summary   | Profiles over time   | Activated destinations  |
|--------|---|--|---|
|        | <p>The audience is defined by the following<br/>pavp - All Female Customer (API)</p> <p>Description</p> <pre>Segments Pandora["label": "Audience", "id": "1", "parent": "1"]   ("Pandora["label": "And", "id": "2", "parent": "1"]     ("Pandora["label": "Older", "id": "3", "parent": "2"]       ("Pandora["label": "Old", "id": "4", "parent": "3"]         ("Pandora["label": "Female", "id": "5", "parent": "4"]           ("Pandora["label": "Female", "id": "6", "parent": "5"]             ("Pandora["label": "Female", "id": "7", "parent": "6"]               ("Pandora["label": "Female", "id": "8", "parent": "7"])))</pre> <p>1 profile<br/>Standard segment</p> | <p>Profiles over time</p>  <p>New audience updated 0 hours ago 24 hours + 0</p> <p>No new audiences qualified for this segment in the selected time range</p> | <p>Activated destinations</p>  <p>You don't have any destinations yet.<br/>Connect to a destination to target your audiences to all your marketing channels.</p> |

- c. Click **Edit segment**. A warning message is displayed, as shown:

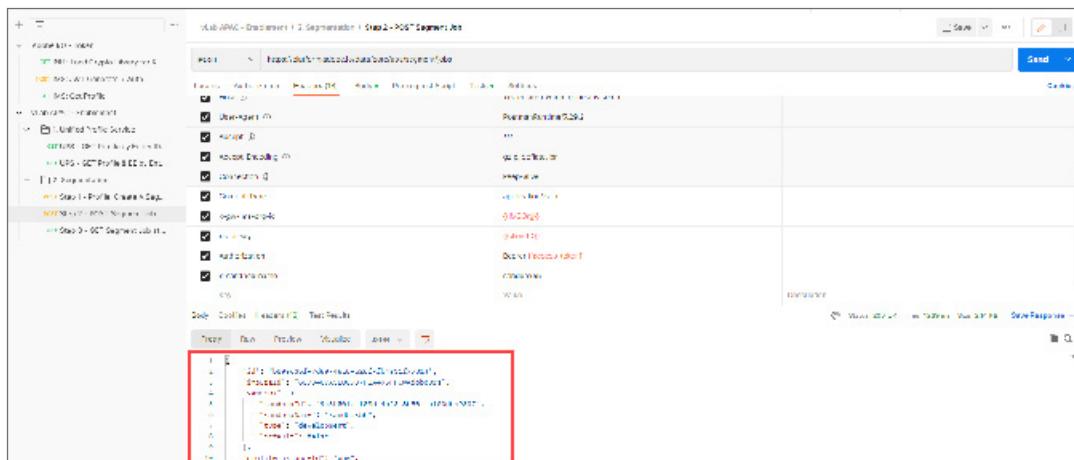


- d. Click **Ok**. A code editor opens.  
e. Click **Cancel** to return to the segment details page.

## Task 2: Post a new segment job request

In the previous exercise, you created definitions for segment. At this moment, no profiles qualify for the segment. Segment jobs process the segment definitions to generate audience segment.

1. Open the **Postman** application and navigate to `_vLab <yourregion> - Enablement >`  
**2. Segmentation > Step 2 - POST Segment Job.** The **POST Step 2 - POST Segment Job** tab opens on the right.
  2. Click the **Headers** tab and add the following details:
    - a. In the **Key** column, type **x-sandbox-name**.
    - b. In the **Value** column, type the sandbox number provided to you by your instructor. For example, **sandbox55**.
  3. Click **Send**. The call is executed and the results are generated, as shown. The segment job is now running. You will check the status of this job in the next task.



## Task 3: Get the segment job request status

1. Open the Postman application and navigate to \_vLab <yourregion> - Enablement > **2. Segmentation > Step 3 - GET Segment Job status.** The **GET Step 3 - GET Segment Job status** tab opens on the right.
2. Click the **Headers** tab and add the following details:
  - a. In the **Key** column, type **x-sandbox-name**.
  - b. In the **Value** column, type the sandbox number provided to you by your instructor. For example, **sandbox55**.
3. Click **Send**. The call is executed and the results are generated, as shown. Notice that the status of the job is displayed as **SUCCEEDED**. In case the job status is **New** or **QUEUED**, repeat the call by clicking **Send** every couple of minutes until the status updates to **PROCESSING** and finally to **SUCCEEDED**.

The screenshot shows the Postman interface with the following details:

- Request URL:** [IP Address]:[Port]/[Path]/[Resource]/segment/job/1024/1024
- Method:** GET
- Headers:**
  - Host: [IP Address]
  - Accept: \*/\*
  - Accept-Encoding: gzip, deflate, br
  - Connection: keep-alive
  - Content-Type: application/json
  - Authorization: [Token ID]
  - x-sandbox-name: sandbox55
- Body (JSON):**

```
{
  "label": "General high level marketing",
  "longitude": "-0.12555555555555555",
  "radius": 1000,
  "sandbox": 1024,
  "segmentName": "Development",
  "type": "Development",
  "version": 1
}
```
- Response:**

```
{
  "id": 1024,
  "label": "General high level marketing",
  "longitude": "-0.12555555555555555",
  "radius": 1000,
  "status": "SUCCEEDED",
  "type": "Development"
}
```

## Module 4

# Querying in Adobe Experience Platform

---

### Introduction

Enterprises manage terabytes of data that come from various sources every day. They struggle to access and analyze the data to provide better customer experiences.

Query service in Adobe Experience Platform enables you to use standard Structured Query Language (SQL) to query your data. You can pull customer datasets into one place and run SQL queries to generate impactful insights from the datasets.

### Objectives

After completing this module, you will be able to:

- Explain Query service in Experience Platform
- Navigate through the Query service user interface (UI) in Experience Platform and execute a query
- Connect PSQL to Experience Platform
- Create queries for the specific XDM values in Experience Platform
- Create queries for the specific XDM values in PSQL
- Create basic queries to analyze data in Experience Platform
- Create basic queries for data analysis in PSQL
- Create queries by using ADF to obtain specific contextual values in Experience Platform
- Create queries in PSQL by using ADF to obtain specific contextual values
- Create a query combining online, call center, and loyalty data available in the Experience Platform UI
- Create a query combining online, call center, and loyalty data available in PSQL
- Create a dataset from the query result in the Experience Platform UI
- Create a visualization report in Power BI Desktop by using the Experience Platform dataset

# Query Service in Experience Platform

Query service in Experience Platform is a serverless tool that enables you to:

- Use standard SQL to query data.
- Join any datasets in Data Lake.
- Capture query results as new datasets. The new datasets can be used in reporting, Data Science Workspace, and Real-time Customer Profile.
- Connect and analyze customer journey across channels.

You can create the SQL queries by using the Query service UI or the Representational State Transfer (RESTful) application programming interface (API).

In the Query service UI, you can:

- Write and execute queries
- View previously executed queries
- Access queries saved within your Integrated Management System (IMS) organization
- Test the queries before executing them on a wider dataset

By using the RESTful API, you can:

- Programmatically write and execute queries
- Schedule queries for future use
- Create templates for queries

You can connect Experience Platform to external query clients, such as Looker, Power BI Desktop, POSTGRESQL (PSQL), and Tableau, to create datasets.

## Adobe-Defined Functions

Query service provides Adobe-defined functions (ADF) that provide advanced functions relevant to Adobe Analytics users. ADFs are prebuilt functions in Query service that help perform common business-related tasks on Experience Event data.

# Exercise 1: Navigate through the Query service UI in Experience Platform and execute a query

This exercise includes the following tasks:

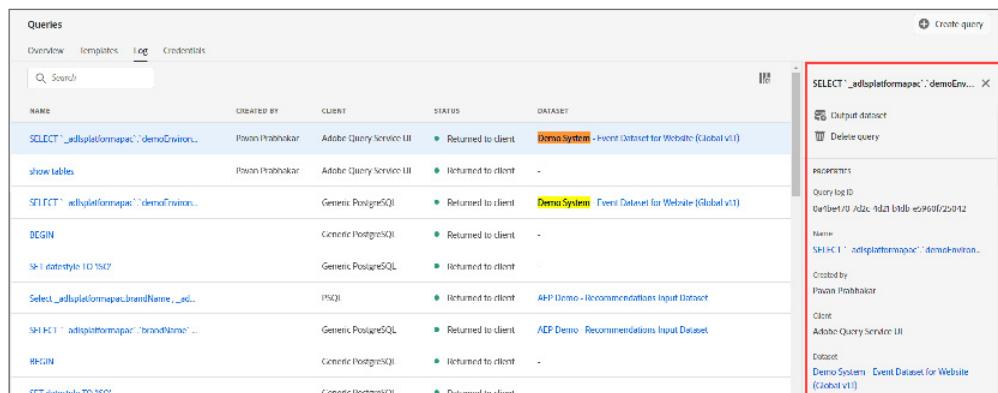
1. Navigate through the Query service UI in Experience Platform
2. Execute a query and view the result

## Task 1: Navigate through the Query Service UI in Experience Platform

1. Open a web browser and type the <https://platform.adobe.com> URL. The **Adobe Experience Platform** page opens.

 **Note:** If prompted, type the credentials provided to you by your Instructor to log in to Experience Platform.

2. Ensure that you are connected to the correct Experience Platform instance.
3. On the left navigation pane, navigate to **DATA MANAGEMENT > Queries**. The **Queries** page opens.
4. Click the **Log** tab. The tab displays the list of queries that were executed in the past.
5. Type **Demo system** in the **Search** box. The search results are displayed.
6. Select a row of an entry. Ensure that you do not click the links in the **NAME** or the **DATASET** columns. A pane opens on the right, as shown:



The screenshot shows the 'Queries' page in the Adobe Experience Platform interface. The 'Log' tab is selected. A search bar contains the text 'Demo system'. A single search result is highlighted in blue, showing the following details:

| Name  | Created By      | Client                 | Status             | Dataset   |
|---|-----------------|------------------------|--------------------|---|
| SELECT * FROM `adlsplatformmapc`.demoEnv... | Pavan Prabhakar | Adobe Query Service UI | Returned to client | Demo System - Event Dataset for Website (Global v1) |

To the right, a detailed view pane is open for the selected query:

- Output dataset:** Demo System - Event Dataset for Website (Global v1)
- PARAMETERS:**
  - Query ID: Q-12345678901234567890123456789012
  - Name: SH-HCI - adlsplatformmapc - demo-env
- Created by:** Pavan Prabhakar
- Client:** Adobe Query Service UI
- Dataset:** Demo System - Event Dataset for Website (Global v1)

7. Scroll down the right pane to view the **SQL QUERY** section.
8. (Optional) Click the **Copy to clipboard** icon, as shown. A message is displayed indicating that the query is copied. Paste the copied query to a text editor.

The screenshot shows the 'Queries' interface in Adobe Experience Platform. On the left, there's a table with columns: NAME, CREATED BY, CLIENT, STATUS, and DATASET. There are four rows listed:

| NAME                                       | CREATED BY       | CLIENT                 | STATUS             | DATASET   |
|--|------------------|------------------------|--------------------|---|
| SELECT `adispplatformapac`.`demoEnviron... | Ritesh Prabhakar | Adobe Query Service UI | Returned to client | Demo System - Event Dataset for Website (Global v1.1) |
| show tables                                | Ritesh Prabhakar | Adobe Query Service UI | Returned to client | -   |
| SELECT `adispplatformapac`.`demoEnviron... |                  | Generic PostgreSQL     | Returned to client | Demo System - Event Dataset for Website (Global v1.1) |

On the right, a large panel displays the SQL code for the first query. A red box highlights the 'Copy to clipboard' icon at the top right of this panel.

```
SELECT `adispplatformapac`.`demoEnv...`
```

SQL DDL/TIFF  
SQL FCT  
`adispplatformapac`.`demoEnvironment.b`  
AS  
`adispplatformapac`.`demoEnvironment.b`  
AS  
`adispplatformapac`.`demoEnvironment.b`  
AS  
`adispplatformapac`.`demoEnvironment.co`  
AS

---

 **Note:** This UI enables you to create new datasets from the executed queries. You can link these datasets to the Adobe Experience Platform Real-Time Customer Profile or use the datasets as the input for your Adobe Experience Platform Data Science Workspace. In SQL terminology, this is similar to a **CREATE TABLE** statement.

---

## Task 2: Execute a query and view the result

- On the **Queries** page, click **Create query**, as shown. The query editor opens.

The screenshot shows the 'Queries' page interface. At the top right, there is a 'Create query' button with a plus sign icon, which is highlighted with a red box. Below it, there are tabs for 'Overview', 'Templates', 'Log', and 'Credentials'. A search bar is present. The main area displays a table with columns: NAME, CREATED BY, CLIENT, STATUS, and DATASET. Two rows are visible: one for 'SELECT ...' and another for 'BEGIN'. On the right side, there is a panel with options like 'Output dataset' and 'Delete query'.

 **Note:** If you are using the query editor in Experience Platform for the first time, you may get a data obligation warning message. Select the **Don't show again** checkbox and click **x** to close the message.

- In the query editor, type **Show tables;**
- Click the play button to run the query or press the Shift+Enter keys. The query results are displayed on the **Results** tab, as shown:

The screenshot shows the query editor with the title 'Queries > Untitled query'. Above the editor, there is a button labeled '(SHIFT + ENTER)'. The editor itself contains the SQL command '1 Show tables;'. Below the editor, there is a results table with two rows. The table has three columns: 'name', 'datasetId', and 'dataset'. The first row has values 'adwh\_dim\_br\_namespace\_destinations', 'c70c6083-752c-4250-aa07-800804192e66', and 'qsaccel'. The second row has values 'adwh\_dim\_br\_segment\_destinations', '792f0a86-4d7c-44d2-ace3-119facf9a055', and 'qsaccel'. The 'Results' tab is selected at the bottom.

| name                                 | datasetId                            | dataset |
|--------------------------------------|--------------------------------------|---------|
| 1 adwh_dim_br_namespace_destinations | c70c6083-752c-4250-aa07-800804192e66 | qsaccel |
| 2 adwh_dim_br_segment_destinations   | 792f0a86-4d7c-44d2-ace3-119facf9a055 | qsaccel |

4. Click the **Console** tab to view the query execution information, as shown:

The screenshot shows the Query Console interface. At the top, it says "Queries > Untitled query". Below that is a toolbar with a play button and the text "(SHIFT + ENTER)". The main area contains a single line of code: "1 Show tables;". Below this is a large blacked-out section. At the bottom, there are two tabs: "Console" (which is selected) and "Results". The "Console" tab displays a log of events:  
11:37:16 AM > Initializing connection...  
11:37:16 AM > Loading autocomplete for tables and columns...  
11:37:16 AM > Connection successfully established.  
11:37:16 AM > Loading autocomplete for tables and columns...  
11:37:17 AM > Autocomplete for tables and columns successfully loaded.  
11:37:18 AM > Autocomplete for tables and columns successfully loaded.  
11:37:44 AM > Running query...  
11:37:44 AM > Query complete in 0.432 seconds. 46 rows returned.

5. Click **Cancel** to return to the **Queries** page.

## (Optional) Exercise 2: Connect PSQL to Experience Platform

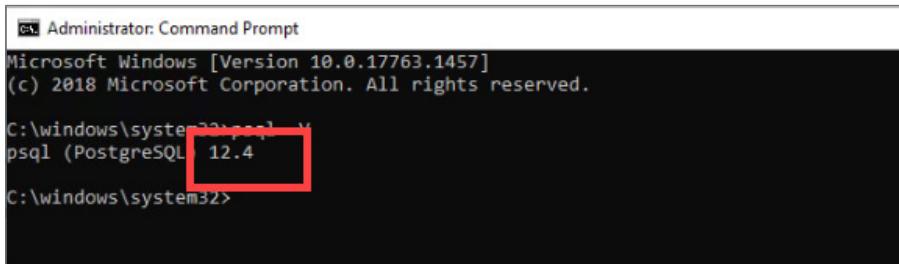
In this exercise, you will establish a connection from Experience Platform to PSQL.

This exercise includes the following tasks:

1. Open Command Prompt and verify the PSQL version
2. Connect PSQL to Experience Platform
3. Execute a query to verify the connection

### Task 1: Open Command Prompt and verify the PSQL version

1. On the Windows search bar, type **Command Prompt**. The search results are displayed.
2. Right-click **Command Prompt** and select **Run as administrator**. The application opens.
3. Type **psql -V** and press the Enter key. The installed PSQL version details are displayed, as shown:



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.1457]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\windows\system32>psql -V
psql (PostgreSQL) 12.4
C:\windows\system32>
```

---

 **Note:** Do not close the **Command Prompt**. You will be switching between **Command Prompt** and the Experience Platform UI to perform the exercises in this module.

---

## Task 2: Connect PSQL to Experience Platform

1. Open Experience Platform and navigate to **DATA MANAGEMENT > Queries** on the left pane. The **Queries** page opens.
  2. Click the **Credentials** tab. The **Postgres credentials** section is displayed.
  3. In the **PSQL command** section, click the **Copy to clipboard** icon, as shown, to copy the PSQL command. The command enables you to connect a PSQL-enabled application to Experience Platform.

| Postgres credentials  |  |
|---|--|
| <a href="#">Host</a>  | <a href="#">psql command ⓘ</a>   |
| <a href="#">adlsplatformapac.platform-query.adobe.io</a>  |   |
| <a href="#">Port</a>  | <a href="#">host=adlsplatformapac.platform-query.adobe.io</a>  |
| 80  | <a href="#">port=80</a>  |
| <a href="#">Database</a>  | <a href="#">dbname=sandbox55:all</a>   |
| <a href="#">sandbox55:all</a>   | <a href="#">user=60794C7A5E8C997F0A495FFC@AdobeOrg</a>   |
| <a href="#">Username</a>  | <a href="#">password=eNqrVkrMSVeYUgoKNjI1U9JRqjAtBfIyc4vj8xINdbNTK3UTS3QW9ZJ14CS2ZkpCsBepkIJUC1x8K1lgc4CRff..enqNUlFT4jaQ_i99bkbsS2rQwczNXXCqegmBrYzDTSZOUC9Km0wREHf_7pYA-6MvtS57fvu1s9tu8w43ZQtI7pW6AQoD2EUzzwvAv51GODI8QAKHU385mDmSeFTTnCYzzsLDvVsrmU3BYRs15m57ovDeqsglc606bm85qRujb0tVHmhAjwQDake_DURD14gsQeT0qe_7viMmcj2NDvproMvP7ylmtzDvlyuvrl9a16rf6YhSORBLmkDtntkpfKs0U7Z3pnDzvLeLmZJBxTcY63azf0nsAUjR6uUmex6AxQj7FzNsJg_pZgZv0VLc3MN4trge3sYPg0myQMwTngu5pe3oxGYJoPRJH6IZo_zFvo-XM-j1fvhCiXXKVhKup3XJfz-R6</a> |
| <a href="#">Password</a>  |   |
| <a href="#">eNqrVkrMSVeYUgoKNjI1U9JRqjAtBfIyc4vj8xINdbNTK3UTS3QW9ZJ14CS2ZkpCsBepkIJUC1x8K1lgc4CRff..enqNUlFT4jaQ_i99bkbsS2rQwczNXXCqegmBrYzDTSZOUC9Km0wREHf_7pYA-6MvtS57fvu1s9tu8w43ZQtI7pW6AQoD2EUzzwvAv51GODI8QAKHU385mDmSeFTTnCYzzsLDvVsrmU3BYRs15m57ovDeqsglc606bm85qRujb0tVHmhAjwQDake_DURD14gsQeT0qe_7viMmcj2NDvproMvP7ylmtzDvlyuvrl9a16rf6YhSORBLmkDtntkpfKs0U7Z3pnDzvLeLmZJBxTcY63azf0nsAUjR6uUmex6AxQj7FzNsJg_pZgZv0VLc3MN4trge3sYPg0myQMwTngu5pe3oxGYJoPRJH6IZo_zFvo-XM-j1fvhCiXXKVhKup3XJfz-R6</a> |   |
| <a href="#">Expires:</a> 10/06/2022 at 8:04 AM  |   |

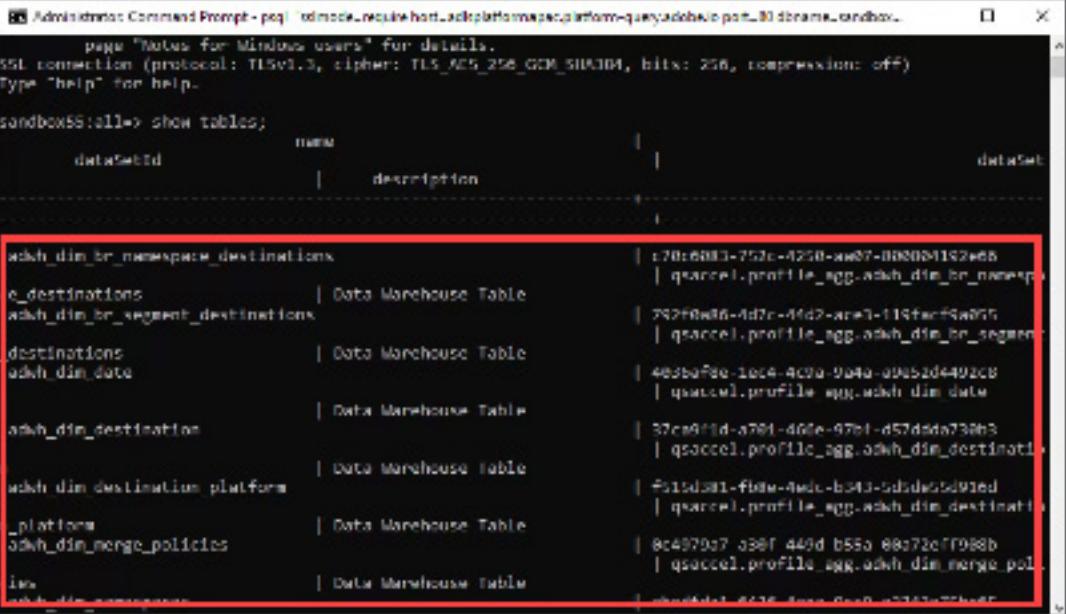
4. Open **Command Prompt**, paste the copied command and press the Enter key. The result is displayed, as shown:

For Mac users: In the OSX terminal window, paste the copied commands from the Experience Platform and press **Return** on the keyboard.

The PSQL client is now connected to the Query Service feature in Experience Platform.

### Task 3: Execute a query to verify the connection

1. In **Command Prompt**, type **Show tables;** and press the Enter key. The results are displayed, as shown:



| datasetId | name                               | description          | dataset                                    |
|-----------|------------------------------------|----------------------|--|
|           | wash_dim_br_namespace_destinations |                      | c70c6003-752c-4250-aa07-000001162e68       |
|           | e_destinations                     | Data Warehouse Table | qsaccel.profile_agg.wash_dim_br_namespace_ |
|           | wash_dim_br_segment_destinations   |                      | 793f9a86-4d7c-44d2-aaa3-119fe79a86         |
|           | destinations                       | Data Warehouse Table | qsaccel.profile_agg.wash_dim_br_segment_   |
|           | adwh_dim_data                      |                      | 4e3bafee-1ec4-4c8a-9040-00052d4492c8       |
|           | adwh_dim_destination               | Data Warehouse Table | 137ca91fd-a781-44de-97e1-d57ddda730b3      |
|           | wash_dim_destination_platform      | Data Warehouse Table | fa15d881-f8a8-4edc-b343-5d5d559595ed       |
|           | platforms                          | Data Warehouse Table | 8c4979a7-00ef-449d-b553-08a72cf7988b       |
|           | adwh_dim_merge_policies            |                      | qsaccel.profile_agg.adwh_dim_merge_pol_    |
|           | ims                                | Data Warehouse Table | 137ca91fd-a781-44de-97e1-d57ddda730b3      |
|           | wash_dim                           |                      | qsaccel.profile_agg.wash_dim_wash_dim_     |

2. Scroll to the end of the table. Notice that **--More--** is displayed at the end of the table.
3. Press Spacebar to view more details.
4. Type **q** in the **Command Prompt** and press enter to view the complete table.

## Exercise 3: Create queries for the specific XDM values in Experience Platform

In this exercise, you will learn how to use queries to retrieve information from the Experience Data Model (XDM) dataset. Every dataset in Experience Platform has a corresponding table that can be referenced by Query service. You can find the table for a dataset on the **Datasets** page.

This exercise includes the following tasks:

1. Open and view the XDM schema
2. Execute queries for the specific XDM values in Experience Platform

### Task 1: Open and view the XDM schema

1. In the Experience Platform's left navigation pane, navigate to **DATA MANAGEMENT > Datasets**. The **Datasets** page opens.
2. Click the **Browse** tab, if not already open. The existing datasets are displayed.
3. In the **Search** box, type **Website**. The search results are displayed.
4. Click the **Demo System - Event Dataset for Website (Global v1.1)** link. The dataset opens, as shown:

The screenshot shows the 'Dataset activity' tab of the 'Demo System - Event Dataset for Website (Global v1.1)' dataset. It displays various metrics: Total records in previous month (802 K), Ingested records in the last 7 days (256), Ingested batches today (0), Ingested batches in the last 7 days (32), Size of data in previous month (218.71 MB), Failed batches today (0), and Failed batches in the last 7 days (0). On the right, there is a sidebar with the dataset's name, a 'Metrics and graphs' section, and a table with dataset details: Name (Demo System - Event Dataset for Website (Global v1.1)), Description (Demo System - Event Dataset for Website...), and Dataset ID (6319e2a51d07311c074d2544).

5. On the right, in the **Table name** section, notice **demo\_system\_event\_dataset\_for\_website\_global\_v1\_1**. Query service will use the tables in the datasets to generate the results of queries.

6. On the right pane, scroll down to view the **Schema** section.

The screenshot shows the 'Dataset activity' tab selected in the top navigation bar. The main area displays various metrics: Total records in previous month (845k), Ingested records in the last 7 days (25), Ingested batches today (2), Failed batches today (2), Ingested batches in the last 7 days (3), Failed batches in the last 7 days (3), and Ingested records (20). To the right, a detailed view of the 'Demo System - Event Dataset for Website (Global v1.1)' is shown. A red box highlights the 'Schema' section, which contains the link 'Demo System - Event Schema for Website (Global v1.1)'. Below this, under 'Source', it says 'Schema'.

7. Click **Demo System - Event Schema for Website (Global v1.1)**. A pop-up window opens, as shown, with the information of the schema.

The screenshot shows the schema details for 'Demo System - Event Schema for Website (Global v1.1)'. A red box highlights the 'Schema name' section, which contains the link 'Demo System - Event Schema for Website (Global v1.1)'. Other visible details include Database ID (6319e2a51d0/311c0/4d25-41), Table name (demo system event dataset for website), Profile (disabled), Schema (Demo System - Event Schema for Website (Global v1.1)), and Created (09/08/2022, 6:10 PM).

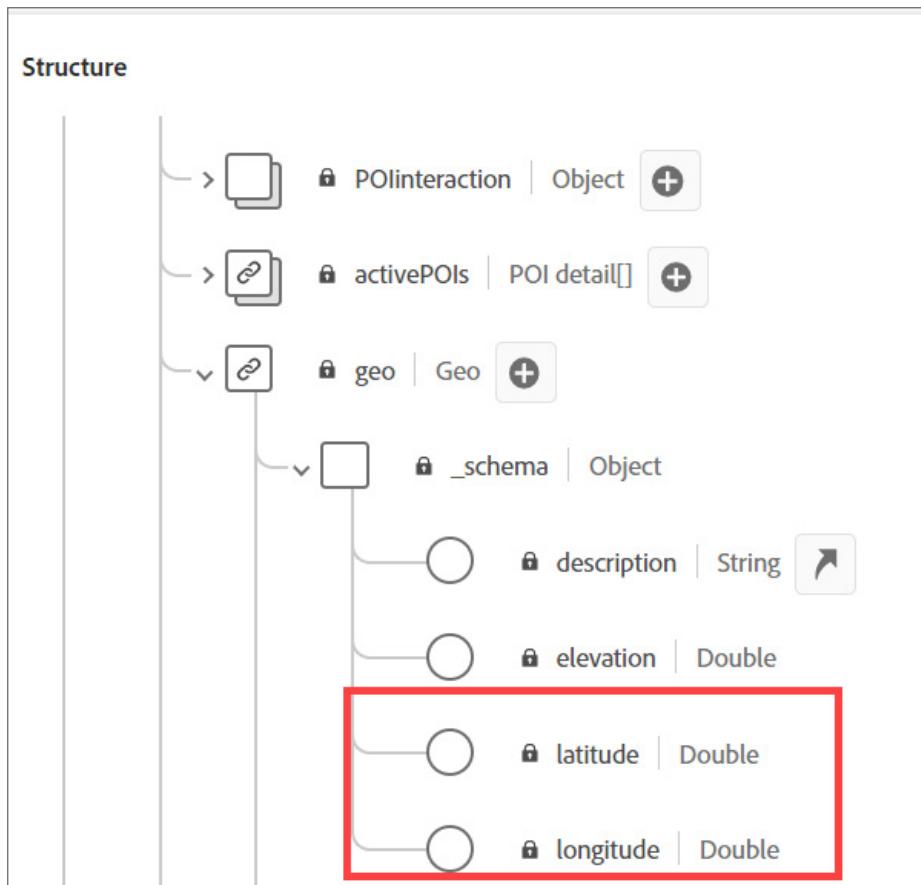
8. Click the link in the **Schema name** section. The **Demo System - Event Schema for Website (Global v1.1)** page opens on a new tab of the browser.

9. In the **Structure** section, scroll down to **placeContext**, as shown, and expand it to view the available fields.

The screenshot shows the AEM Schema Structure interface. On the left, there's a sidebar with 'Composition' and 'Schema' sections. Under 'Schema', there's a list of schemas including 'Demo System - Event Schema for Website (Global v1.1)'. The main area is titled 'Structure' and contains a tree view of schema nodes. One node, 'placeContext', is highlighted with a red box. Other visible nodes include 'adspplatformapac', 'experience', 'commerce', 'device', 'environment', 'ImplementationDetails', 'marketing', 'productListItems', and 'search'. Each node has a small icon, a lock symbol indicating security, and a plus sign for further expansion.

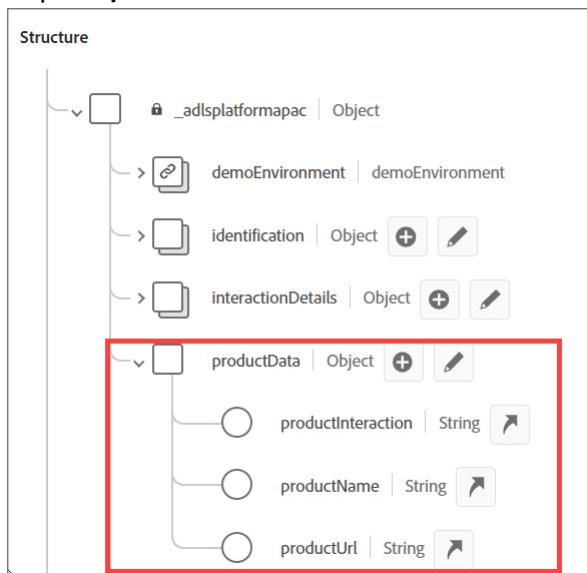
10. Expand **geo** and notice the available fields such as **city** and **countryCode**.

11. Expand **\_schema** under **geo** to view the available fields such as the **latitude** and **longitude** fields, as shown:



12. Scroll up and expand **\_adlsplatform<yourregion>**. For example, **\_adlsplatformapac**. The available fields are displayed.

13. Expand **productData** and notice the available fields, as shown:



14. Click **Cancel** to return to the **Schemas** page.

## Task 2: Execute queries for the specific XDM values in Experience Platform

To query the location for product view, you need to select **geo** and **productInteraction**. You will use the dot (.) notation to navigate through XDM.

1. On the left navigation pane, navigate to **DATA MANAGEMENT > Queries**. The **Queries** page opens.
2. Ensure you are in the **Browse** tab and click **Create query** at the upper-right corner to open the query editor.
3. From the exercise folder provided to you, open the **CodeFile\_M4\_Ex3.txt** file, copy the SQL statement in the **Module 4 > Exercise 3 > Task 2 > Step 3** section, and paste the statement in the query editor.
4. Click the play button or press the Shift+Enter keys to run the query. The results are displayed, as shown:

The screenshot shows the 'Queries > Untitled query' interface. In the 'Console' tab, a SQL query is pasted and executed. The results are shown in the 'Results' tab, where a single row is returned with the value '(57.4694803,-3.1269422)"Tullich,GB)'.

```
1 select placeContext.geo
2 from aep_demo_luma_telco_website_interactions_dataset
3 where _adlsplatformapac.productData.productInteraction = 'productView'
4 and placeContext.geo.countryCode <> ''
5 limit 1;
```

| geo                                    |
|--|
| 1 ("57.4694803,-3.1269422)"Tullich,GB) |

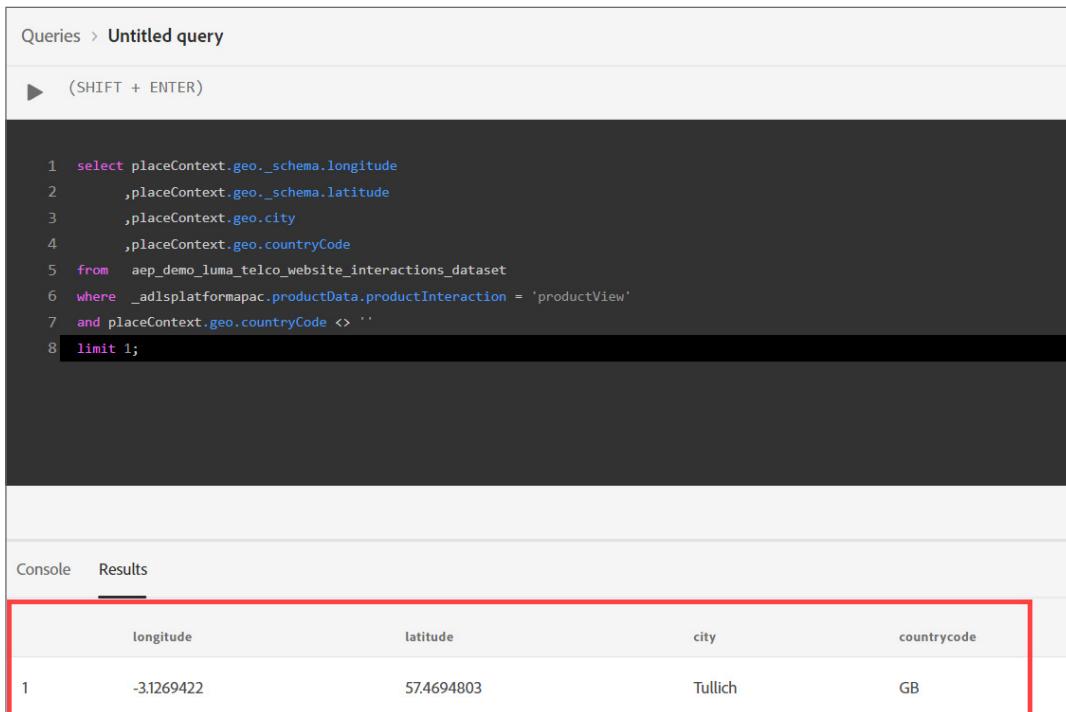
5. Observe that the results of the query are a group of values such as ("57.4694803,-3.1269422","Tullich,GB) and not an individual value.

---

**Note:** Observe that the result is a flattened object rather than a single value. In some systems, this is referred to as a vector of values. In schema, the **placeContext** object has numerous fields defined. However, the **placeContext.geo** object in the dataset contains the values for the following three attributes: **schema** (longitude and latitude), **country**, and **city**. When an object is declared as column in a query, it returns the entire object.

---

6. In the query editor, delete the existing query.
7. Open the **CodeFile\_M4\_Ex3.txt** file if not open already, copy the SQL statement in the **Module 4 > Exercise 3 > Task 2 > Step 6** section, and paste the statement in the query editor. To select the individual values of an object, you will use the dot (.) notation in the query.
8. Click the play button to execute the query. The query results are displayed, as shown:



```
Queries > Untitled query

▶ (SHIFT + ENTER)

1 select placeContext.geo._schema.longitude
2 ,placeContext.geo._schema.latitude
3 ,placeContext.geo.city
4 ,placeContext.geo.countryCode
5 from aep_demo_luma_telco_website_interactions_dataset
6 where _adlsplatformformapac.productData.productInteraction = 'productView'
7 and placeContext.geo.countryCode <> ''
8 limit 1;

Console Results
longitude latitude city countrycode
1 -3.1269422 57.4694803 Tullich GB
```

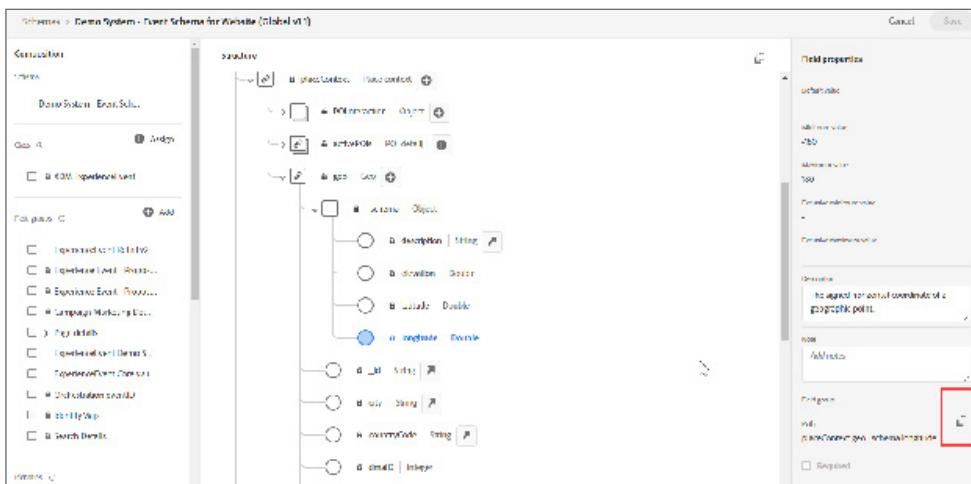
Notice that the result is a set of individual and scalar values.

9. Click **Cancel** to return to the **Queries** page.
10. Open a text editor. For this exercise, open **Notepad**.

11. From the **CodeFile\_M4\_Ex3.txt** file, copy the SQL statement in the **Module 4 > Exercise 3 > Task 2 > Step 10** section, and paste it in the text editor, as shown:

```
select <your_attribute_path_here>
from aep_demo_luma_telco_website_interactions_dataset
where _adlsplatformapac.productData.productInteraction = 'productView'
and placeContext.geo.countryCode <> ''
limit 1;
```

12. In Experience Platform, navigate to **DATA MANAGEMENT > Schemas** on the left navigation pane. The **Schemas** page opens.
13. Click the **Browse** tab. A list of existing schemas opens.
14. In the **Search** box, type **website** to get a list of schemas.
15. From the list, click **Demo System - Event Schema for Website (Global v1.1)**. The schema structure and details open.
16. In the **Structure** section, scroll down and expand **placeContext**. The available fields are displayed.
17. Expand **geo**. The available fields are displayed.
18. Expand **\_schema**. Notice the **latitude** and **longitude** fields under schema.
19. Select **longitude**. The **Field properties** pane is displayed on the right.
20. On the right pane, scroll down to view the **Path** section and click the **Copy path** icon, as shown, to copy the path. A message is displayed indicating that the path is copied.



21. Open the text editor and delete <your\_attribute\_path\_here> from the first line, as shown:

```
select <your_attribute_path_here>
from aep_demo_luma_telco_website_interactions_dataset
where _adlsplatformapac.productData.productInteraction = 'productView'
and placeContext.geo.countryCode <> ''
limit 1;
```

22. Paste the copied path from Experience Platform, as shown:

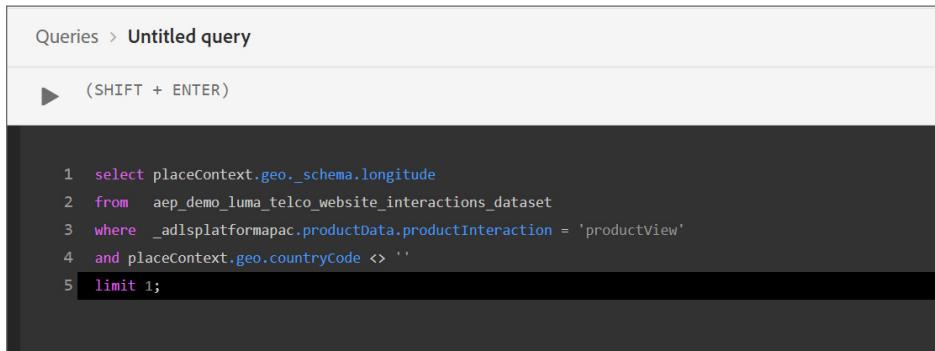
```
select placeContext.geo._schema.longitude
from aep_demo_luma_telco_website_interactions_dataset
where _adlsplatformapac.productData.productInteraction = 'productView'
and placeContext.geo.countryCode <> ''
limit 1;
```

23. Copy the updated code from the text editor.

24. Open Experience Platform and navigate to **DATA MANAGEMENT > Queries**. The **Queries** page opens.

25. Click **Create query**. The query editor opens.

26. Paste the code in the query editor, as shown:



27. Click the play button to execute the query. The result is displayed, as shown:

The screenshot shows a query editor interface. At the top, it says "Queries > Untitled query". Below that is a button labeled "▶ (SHIFT + ENTER)". The main area contains the following SQL code:

```
1 select placeContext.geo._schema.longitude
2 from aep_demo_luma_telco_website_interactions_dataset
3 where _adlsplatformapac.productData.productInteraction = 'productView'
4 and placeContext.geo.countryCode <> ''
5 limit 1;
```

Below the code, there are two tabs: "Console" and "Results". The "Results" tab is selected, showing a single row of data:

|   | longitude  |
|---|------------|
| 1 | -3.1269422 |



**Note:** The **from** and **where** clauses in the query and the value for longitude will be different for you depending on the training instance.

## (Optional) Exercise 4: Create queries for the specific XDM values in PSQL

In this exercise, you will execute the same queries that you executed in Exercise 3, but you will use PSQL instead of Query service UI to execute the queries.

1. Open the **CodeFile\_M4\_Ex4.txt** file and copy the code in the **Module 4 > Exercise 4 > Step 1** section.
2. Open **Command Prompt** and paste the copied code, as shown:

```
sandbox55:all-> select placeContext.geo  
sandbox55:all-> from aep_demo_luma_telco_website_interactions_dataset  
sandbox55:all-> where _adlplatformmapac.productData.productInteraction = 'productView'  
sandbox55:all-> and placeContext.geo.countryCode <> ''  
sandbox55:all-> limit 1;
```

3. Press the Enter key to execute the query. The result is displayed, as shown:

```
sandbox55:all-> select placeContext.geo  
sandbox55:all-> from aep_demo_luma_telco_website_interactions_dataset  
sandbox55:all-> where _adlplatformmapac.productData.productInteraction = 'productView'  
sandbox55:all-> and placeContext.geo.countryCode <> ''  
sandbox55:all-> limit 1;  
geo  
((57.4694889,-3.1209422),Tulllich,GB)  
1 row
```

4. From the **CodeFile\_M4\_Ex4.txt** file, copy the code in the **Module 4 > Exercise 4 > Step 4** section.
5. Open **Command Prompt**, paste the copied code and press the Enter key. The result is displayed, as shown:

```
sandbox55:all-> select placeContext.geo.schema.longitude  
sandbox55:all-> ,placeContext.geo.schema.latitude  
sandbox55:all-> ,placeContext.geo.city  
sandbox55:all-> ,placeContext.geo.countryCode  
sandbox55:all-> from aep_demo_luma_telco_website_interactions_dataset  
sandbox55:all-> where _adlplatformmapac.productData.productInteraction = 'productView'  
sandbox55:all-> and placeContext.geo.countryCode <> ''  
sandbox55:all-> limit 1;  
longitude | latitude | city | countryCode  
-3.1209422 | 57.4694889 | Tulllich | GB  
1 row
```

6. Open a text editor. For example, open **Notepad**.
  7. Open the **CodeFile\_M4\_Ex4.txt** file and copy the code in the **Module 4 > Exercise 4 > Step 7** section.
  8. Paste the code in the text editor.
  9. Open Experience Platform and navigate to **DATA MANAGEMENT > Schemas**. The **Schemas** page opens.
  10. Click the **Browse** tab. A list of existing schemas is displayed.
  11. In the **Search field** box, type **website**. The search results are displayed.
  12. Click **Demo System - Event Schema for Website (Global v1.1)**. The schema structure and details are displayed.
  13. In the **Structure** section, expand **placeContext**. The available fields are displayed.
  14. Expand **geo**. The available fields are displayed.
  15. Expand **\_schema**. The available fields are displayed.
  16. Select **longitude**. The **Field properties** pane is displayed on the right.
  17. Scroll down to view the **Path** section and click the **Copy path** icon to copy the path. A message is displayed indicating that the path is copied.
  18. Open the text editor and delete **<your\_attribute\_path\_here>** from the first line.
  19. Paste the copied path from Experience Platform, as shown:

```
File Edit Format View Help
select placeContext.geo._schema.longitude
from _prod_level1_product_interactions_dataset
where _adlsplatformapac.productData.productInteraction = 'productView'
and placeContext.geo.countryCode <> ''
limit 1;
```

20. Copy the updated code from the text editor.
  21. Open **Command Prompt** and paste the updated code, as shown:

22. Press the Enter key to execute the query. The result is displayed, as shown:

```
sandbox55:all=>
sandbox55:all=> select placeContext.geo._schema.longitude
sandbox55:all-> from aep_demo_luma_telco_website_interactions_dataset
sandbox55:all-> where _adlsplatformapac.productData.productInteraction = 'productView'
sandbox55:all-> and placeContext.geo.countryCode <> ''
sandbox55:all-> limit 1;
longitude
-----
-3.1269422
(1 row)
```

## Exercise 5: Create basic queries to analyze data in Experience Platform

You need to create a few basic queries to analyze product views, product interaction funnels, and churn. This exercise includes the following tasks:

1. Create a query to obtain the timestamp data
2. Create a query to obtain the top five products viewed
3. Create a query to obtain the product interaction funnel
4. Create a query to identify the visitors with a risk to churn based on the webpage visits

### Task 1: Create a query to obtain the timestamp data

In this task, you will create a query to analyze a data over a period of time such as the product views each day with the number of days limited to 10.

1. In the Experience Platform's left navigation pane, click **Queries**. The **Queries** page opens.
2. Click **Create query** to open the query editor.
3. Open the **CodeFile\_M4\_Ex5.txt** file if not open already and copy the code in the **Module 4 > Exercise 5 > Task 1 > Step 3** section.
4. Open Experience Platform and paste the code in the query editor, as shown:

```
Queries > Untitled query

▶ (SHIFT + ENTER)

1 select date_format( timestamp , 'yyyy-MM-dd') AS Day,
2       count(*) AS productViews
3   from aep_demo_luma_telco_website_interactions_dataset
4  where _adlsplatformapac.demoEnvironment.brandName like 'Luma Telco'
5    and _adlsplatformapac.productData.productInteraction = 'productView'
6 group by Day
7 limit 10;
8
```

5. Click the play button to execute the query. The result is displayed, as shown:

The screenshot shows a query editor window. At the top, there is a play button icon followed by the text "(SHIFT + ENTER)". Below this is a code editor containing the following SQL query:

```
1 select date_format( timestamp , 'yyyy-MM-dd') AS Day,
2       count(*) AS productViews
3   from aep_demo_luma_telco_website_interactions_dataset
4  where _adlsplatformapac.demoEnvironment.brandName like 'Luma Telco'
5    and _adlsplatformapac.productData.productInteraction = 'productView'
6 group by Day
7 limit 10;
8
```

Below the code editor, there are two tabs: "Console" and "Results". The "Results" tab is selected, displaying a table with two columns: "Day" and "productViews". The data is as follows:

|   | Day        | productViews |
|---|------------|--------------|
| 1 | 2022-07-31 | 4594         |



**Note:** Alternatively, you can press the Shift+Enter keys to execute the query.

## Task 2: Create a query to obtain the top five products viewed

1. In the query editor, delete the existing query.
2. Open the **CodeFile\_M4\_Ex5.txt** file and copy the code in the **Module 4 > Exercise 5 > Task 2 > Step 2** section.
3. Open Experience Platform and paste the copied code in the query editor.
4. Click the play button to execute the query. The result is displayed, as shown:

The screenshot shows the Experience Platform Query Editor interface. At the top, there is a toolbar with a play button and the text "(SHIFT + ENTER)". Below the toolbar is a code editor containing a SQL query. The query selects the product name and count from the \_adlsplatformapac.productData table where the brand name is 'Luma Telco' and the interaction type is 'productView', ordered by count in descending order, and limited to the top 5 results. The results are displayed in a table below the code editor. The table has two columns: 'productname' and 'count(1)'. The results are as follows:

|   | productname                           | count(1) |
|---|---------------------------------------|----------|
| 1 | Google Pixel XL 32GB Black Smartphone | 1876     |
| 2 | SIM Only                              | 964      |
| 3 | Samsung Galaxy S8                     | 912      |

### Task 3: Create a query to obtain the product interaction funnel

1. In the query editor, delete the existing query.
2. Open the **CodeFile\_M4\_Ex5.txt** file and copy the code in the **Module 4 > Exercise 5 > Task 3 > Step 2** section.
3. Open Experience Platform and paste the copied code in the query editor.
4. Click the play button to execute the query. The result is displayed, as shown:

The screenshot shows the Experience Platform Query Editor interface. At the top, it says "Queries > Untitled query". Below that is a toolbar with a play button and the text "(SHIFT + ENTER)". The main area contains the following SQL code:

```
1 select adlspplatformapac.productData.productInteraction, count(*)
2 from aep_demo_luma_telco_website_interactions_dataset
3 where adlspplatformapac.environment.brandName like 'luma_telco'
4 and adlspplatformapac.productData.productInteraction <> ''
5 group by adlspplatformapac.productData.productInteraction;
6
```

Below the code, there are tabs for "Console" and "Results". The "Results" tab is selected and displays a table with three rows:

| productinteraction | count(1) |
|--------------------|----------|
| 1 productView      | 4594     |
| 2 productAddToCart | 988      |
| 3 productPurchase  | 492      |

Task 4: Create a query to identify the visitors with a risk to churn based on the webpage visits

1. In the query editor, delete the existing query.
2. Open the **CodeFile\_M4\_Ex5.txt** file and copy the code in the **Module 4 > Exercise 5 > Task 4 > Step 2** section.
3. Open Experience Platform and paste the copied code in the query editor.
4. Click the play button to execute the query. The result is displayed, as shown:

The screenshot shows the Experience Platform Query Editor interface. At the top, there is a toolbar with a play button and the text "(SHIFT + ENTER)". Below the toolbar is a dark code editor area containing a SQL query. The query selects distinct ECIDs from the 'aep\_demo\_luma\_telco\_website\_interactions\_dataset' where the brand name is 'Luma Telco' and the web page details name is 'Cancel Service', grouped by ECID, and limited to 10 rows. The code is numbered from 1 to 7. Below the code editor is a navigation bar with 'Console' and 'Results' tabs, with 'Results' being the active tab. The results table has a single column named 'ecid'. Three rows of data are shown, each with a red border around it. The first row contains the value '67802232253493573025911610627278'. The second row contains '27147331741697745713411940873426'. The third row contains '19806347932758146991274525406147'.

| ecid                               |
|------------------------------------|
| 1 67802232253493573025911610627278 |
| 2 27147331741697745713411940873426 |
| 3 19806347932758146991274525406147 |

## (Optional) Exercise 6: Create basic queries for data analysis in PSQL

In this exercise, you will create basic queries in PSQL instead of the Experience Platform UI. This exercise includes the following tasks:

1. Create a query to obtain the timestamp data
2. Create a query to obtain the top five products viewed
3. Create a query to obtain the product interaction funnel
4. Create a query to identify the visitors with a risk to churn based on the webpage visits

### Task 1: Create a query to obtain the timestamp data

1. Open the **CodeFile\_M4\_Ex6.txt** file and copy the code in the **Module 4 > Exercise 6 > Task 1 > Step 1** section.
2. Open **Command Prompt** and paste the copied code.:.
3. Press the Enter key to execute the query. The result is displayed, as shown:

```
sandbox$5:all-> select date_format(timestamp, 'yyyy-MM-dd') AS Day,
sandbox$5:all->      count(*) AS productViews
sandbox$5:all-> from acp_demo_lumo_telco_website_interactions_dataset
sandbox$5:all-> where _adlplatformspec демонстрация.brandName like 'Lumo Telco'
sandbox$5:all-> and _adlplatformspec.productData.productInteraction = "productView"
sandbox$5:all-> group by Day
sandbox$5:all-> limit 10-


| Day        | productViews |
|------------|--------------|
| 2023-07-31 | 4594         |


(1 row)
```

## Task 2: Create a query to obtain the top five products viewed

1. Open the **CodeFile\_M4\_Ex6.txt** file and copy the code in the **Module 4 > Exercise 6 > Task 2 > Step 1** section.
2. Open **Command Prompt** and paste the copied code.
3. Press the Enter key to execute the query. The result is displayed, as shown:

```
sandbox55:all-> select _adlplatformapac.productData.productName, count(*)  
sandbox55:all-> from _adp_demo_luma_technology_website_interactions_dataset  
sandbox55:all-> where _adlplatformapac.demoEnvironment.brandName like 'Luma Telco'  
sandbox55:all-> and _adlplatformapac.productData.productInteraction = 'productview'  
sandbox55:all-> group by _adlplatformapac.productData.productId, productName  
sandbox55:all-> order by 2 desc  
sandbox55:all-> limit 5;  


| productname                                      | count(*) |
|--------------------------------------------------|----------|
| Google Pixel XL 32GB Black Smartphone   SIM Only | 1876     |
| Samsung Galaxy S8                                | 964      |
| Samsung Galaxy S7 32GB Black (4 GB RAM)          | 912      |
|                                                  | 842      |


```

### Task 3: Create a query to obtain the product interaction funnel

1. Open the **CodeFile\_M4\_Ex6.txt** file and copy the code in the **Module 4 > Exercise 6 > Task 3 > Step 1** section.
2. Open **Command Prompt** and paste the copied code.
3. Press the Enter key to execute the query. The result is displayed, as shown:

```
 sandbox55:all-> select _adisplatformmpc.productData.productInteraction, count(*)  
 sandbox55:all-> from _asp_demo_luma_telco_website_interactions dataset  
 sandbox55:all-> where _adisplatformmpc.demutEnvironment.brandName like 'luma telco'  
 sandbox55:all-> and _adisplatformmpc.productData.productInteraction >> ''  
 sandbox55:all-> group by _adisplatformmpc.productData.productInteraction;  
 productInteraction | count(1)  
  
 productView | 4594  
 productAddToCart | 988  
 productPurchase | 492  
 (3 rows)
```

Task 4: Create a query to identify the visitors with a risk to churn based on the webpage visits

1. Open the **CodeFile\_M4\_Ex6.txt** file and copy the code in the **Module 4 > Exercise 6 > Task 4 > Step 1** section.
2. Open **Command Prompt** and paste the copied code.
3. Press the Enter key to execute the query. The result is displayed, as shown:

```
sandbox55:all-> select distinct _adlsplatformapac.identification.core.ecid  
sandbox55:all-> from _xep_demo_luma_health_website_interactions_dataset  
sandbox55:all-> where _adlsplatformapac.demoEnvironment.brandName like 'Luma Telco'  
sandbox55:all-> and web.webFogcDetails.name = 'Concierge Service'  
sandbox55:all-> group by _adlsplatformapac.identification.core.ecid  
sandbox55:all-> limit 10;  
ecid  
  
07882253253405573825011610627278  
201033431781164710%1141141KK/482h  
19880347933758140991374535486147  
86333676267512351981624626488225  
230333446748775575/816589/66564409  
1188602811340207991112795192858890  
84258867338433466987489111372169  
00257333876958402281324714165751  
00695181015487520438257051973742  
1910385255844885/804345755/80469b  
10 rows)
```

## Exercise 7: Create queries by using ADF to obtain specific contextual values in Experience Platform

---

This exercise includes the following tasks:

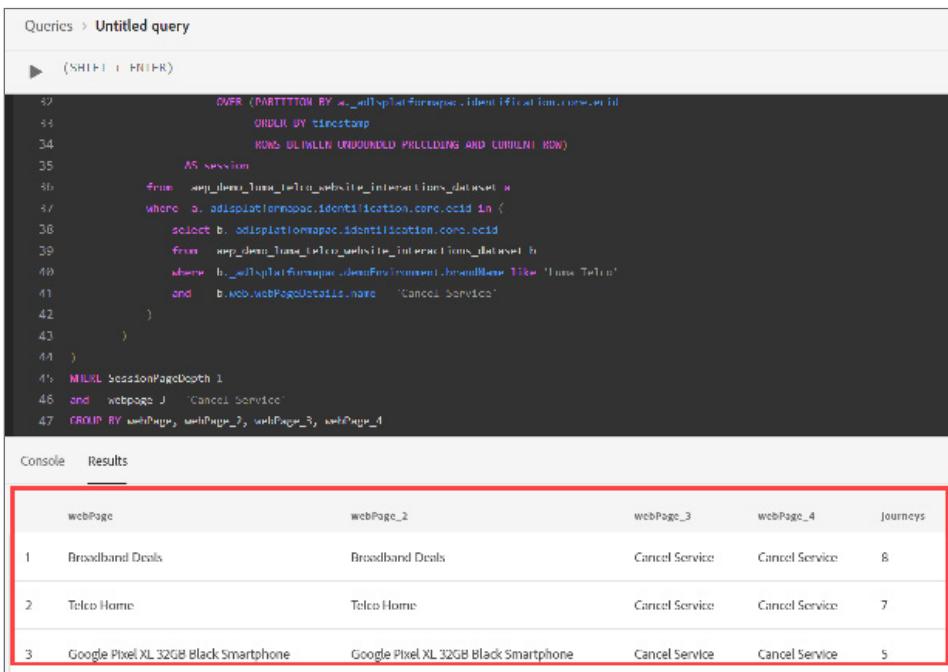
1. Create a query to obtain the customer actions data on the website before reaching the cancel service page as the third page in a session
2. Create a query to obtain the time lapse data before a visitor calls the call center after visiting the cancel service page
3. Create a query to obtain the outcome of the call center interaction
4. Create a query to obtain the loyalty profile of customers
5. Create a query to obtain the geolocation data of the customers visiting the website

**Task 1:** Create a query to obtain the customer actions data on the website before reaching the cancel service page as the third page in a session

To create this query, you will use the following two ADFs:

- **SESS\_TIMEOUT()**: generates the visit groupings found with Adobe Analytics. It performs a similar time-based grouping but with customizable parameters.
  - **NEXT()** and **PREVIOUS()**: helps you understand how customers navigate through your website.
1. In the query editor, delete the existing query.
  2. Open the **CodeFile\_M4\_Ex7.txt** file and copy the code in the **Module 4 > Exercise 7 > Task 1 > Step 2** section.
  3. Open Experience Platform and paste the copied code in the query editor.

4. Click the play button to execute the query. The result is displayed, as shown:



The screenshot shows the AEM Query UI interface. At the top, there's a header bar with 'Queries > Untitled query'. Below it is a code editor containing a complex SQL-like query. The code includes several JOIN clauses, WHERE conditions, and a GROUP BY clause. The results are displayed in a table below the code editor. The table has columns: 'webPage', 'webPage\_2', 'webPage\_3', 'webPage\_4', and 'Journeys'. There are three rows of data, each with a red border around it. Row 1: 'Broadband Deals' in 'webPage', 'Broadband Deals' in 'webPage\_2', 'Cancel Service' in 'webPage\_3', 'Cancel Service' in 'webPage\_4', and '8' in 'Journeys'. Row 2: 'Tele Home' in 'webPage', 'Tele Home' in 'webPage\_2', 'Cancel Service' in 'webPage\_3', 'Cancel Service' in 'webPage\_4', and '7' in 'Journeys'. Row 3: 'Google Pixel XL 32GB Black Smartphone' in 'webPage', 'Google Pixel XL 32GB Black Smartphone' in 'webPage\_2', 'Cancel Service' in 'webPage\_3', 'Cancel Service' in 'webPage\_4', and '5' in 'Journeys'.

|   | webPage                               | webPage_2                             | webPage_3      | webPage_4      | Journeys |
|---|---------------------------------------|---------------------------------------|----------------|----------------|----------|
| 1 | Broadband Deals                       | Broadband Deals                       | Cancel Service | Cancel Service | 8        |
| 2 | Tele Home                             | Tele Home                             | Cancel Service | Cancel Service | 7        |
| 3 | Google Pixel XL 32GB Black Smartphone | Google Pixel XL 32GB Black Smartphone | Cancel Service | Cancel Service | 5        |

Task 2: Create a query to obtain the time lapse data before a visitor calls the call center after visiting the cancel service page

To create this query, you will use the **TIME\_BETWEEN\_NEXT\_MATCH()** function. The time between the previous or the next match functions provide a new dimension that measures the time that elapsed since an incident.

1. In the query editor, delete the existing query.
2. Open the **CodeFile\_M4\_Ex7.txt** file and copy the code in the **Module 4 > Exercise 7 > Task 2 > Step 2** section.
3. Open Experience Platform and paste the copied code in the query editor.

4. Click the play button to execute the query. The result is displayed, as shown:

The screenshot shows the 'Queries > Untitled query' interface. The query code is as follows:

```
1 select * from (
2     select _fullSplitFromQuery.visitId, visition.cookieId as cookieId,
3            web.webPageDetails.name as webPage,
4            LTRIM(RTRIM(MATCH(timestamp, web.webPageDetails.name 'Call Start', 'seconds')
5            OVER(PARTITION BY _fullSplitFromQuery.visitId visition.cookieId
6                  ORDER BY timestamp
7                  ROWS UNLIMITED CURRENT ROW AND UNBOUNDED FOLLOWING))
8            AS contact_collicenter_after_seconds
9    from sep_demo_time_call_to_website_interactions_dataset
10   where _fullSplitFromQuery.deviceEnvironment.browsers like 'Time Telco'
11   and web.webPageDetails.name in ('Cancel Service', 'Call Start')
12 ) p
13 where p.webPage = 'Cancel Service'
14 limit 15;
```

The results table has two columns: 'Console' and 'Results'. The 'Results' table contains four rows, each with a row number (6, 7, 8, 9) and a list of values. The last three rows (7, 8, 9) are highlighted with a red border.

| Console | Results  |
|---------|--|
| 6       | 006304706655544176/99669244202779<br>Cancel Service -797 |
| 7       | 00720075344152/96154458668700428<br>Cancel Service -519  |
| 8       | 00720075344152/96154458668700428<br>Cancel Service -519  |
| 9       | 00746064605049656090779523644276<br>Cancel Service -62   |

### Task 3: Create a query to obtain the outcome of the call center interaction

In this query, you will join the aep\_demo\_website\_interactions and aep\_demo\_call\_center\_interactions datasets to analyze the outcome of the call center interaction.

1. In the query editor, delete the existing query.
2. Open the **CodeFile\_M4\_Ex7.txt** file and copy the code in the **Module 4 > Exercise 7 > Task 3 > Step 2** section.
3. Open Experience Platform and paste the copied code in the query editor.
4. Click the play button to execute the query. The result is displayed, as shown:

The screenshot shows the Experience Platform Query Editor interface. The top part displays the query code in the 'Console' tab, which includes joins between 'aep\_demo\_call\_center\_interactions' and 'aep\_demo\_website\_interactions' datasets, filtering for 'Cancel Service' calls from Luxe Telco, and ordering by timestamp. The bottom part shows the 'Results' tab with a table containing three rows of data. The columns are labeled: ecid, webpage, contact\_callcenter\_after\_seconds, callfeeling, calltopic, and contractcancelled. The data rows are:

|   | ecid                             | webpage        | contact_callcenter_after_seconds | callfeeling | calltopic | contractcancelled |
|---|----------------------------------|----------------|----------------------------------|-------------|-----------|-------------------|
| 1 | 00331886620679939148047665693117 | Cancel Service |                                  | none        | none      | no                |
| 2 | 00331886620679939148047665693117 | Cancel Service |                                  | none        | none      | no                |
| 3 | 00626561600197295702131319716066 | Cancel Service |                                  | none        | none      | no                |

## Task 4: Create a query to obtain the loyalty profile of customers

In this query, you will join the loyalty data that was onboarded into Experience Platform. This enables you to enrich the churn analysis with loyalty data.

1. In the query editor, delete the existing query.
2. Open the **CodeFile\_M4\_Ex7.txt** file and copy the code in the **Module 4 > Exercise 7 > Task 4 > Step 2** section.
3. Open Experience Platform and paste the copied code in the query editor.
4. Click the play button to execute the query. The result is displayed, as shown:

```
6  from (
7      select adispIotionnapoc.identification.core.acid,
8         Web.webPageDetails.name as webPage,
9         TINR_BFTMFFW_NEXT_MATCH(timestamp, web.webPageDetails.name='Call Start', 'seconds')
10        OVER(PARTITION BY _adispIotionnapoc.identification.core.acid
11          ORDER BY timestamp
12          ROW UNLIMITED PRECEDING AND UNBOUNDED FOLLOWING)
13          AS contact_center_after_seconds
14    from xep_demo_time_to_call_center_interactions_dataset
15   where adispIotionnapoc.demoEnvironment.brandName like 'Luna Telco'
16   and Web.webPageDetails.name in ('Cancel Service', 'Call Start')
17 ) e
18 , xep_demo_time_to_call_center_interactions_dataset x
19 , demo system profile dataset for crn global v1 1 1
20 where e.acid = x._adispIotionnapoc.identification.core.acid
21 and x.webPage = 'Cancel Service'
```

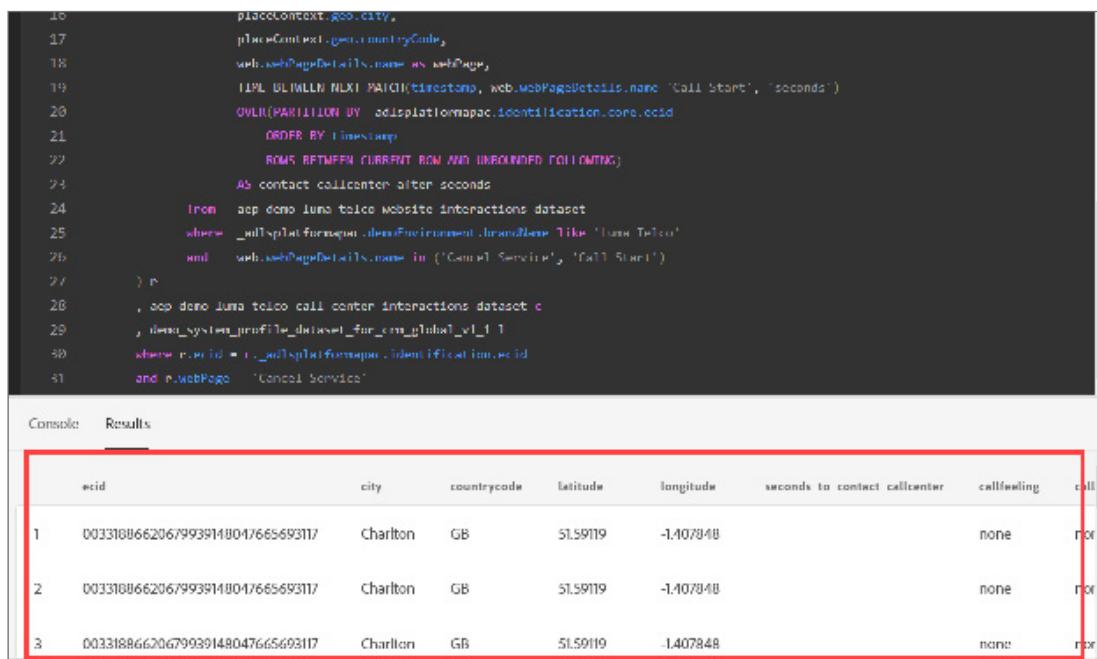
Console    Results

| acid | webPage                          | contact_center_after_seconds | callfeeling | calltopic | level  | crnid     |
|------|----------------------------------|------------------------------|-------------|-----------|--------|-----------|
| 1    | 00331886620679939148047665693117 | Cancel Service               | none        | none      | Bronze | 267095232 |
| 2    | 00331886620679939148047665693117 | Cancel Service               | none        | none      | Bronze | 267095232 |
| 3    | 00331886620679939148047665693117 | Cancel Service               | none        | none      | Bronze | 267095232 |

## Task 5: Create a query to obtain the geolocation data of the customers visiting the website

You will include the geographical information, such as longitude, latitude, city, and countryCode, captured by Experience Platform to get the geographical insights about churning customers.

1. In the query editor, delete the existing query.
2. Open the **CodeFile\_M4\_Ex7.txt** file and copy the code in the **Module 4 > Exercise 7 > Task 5 > Step 2** section.
3. Open Experience Platform and paste the copied code in the query editor.
4. Click the play button to execute the query. The result is displayed, as shown:



The screenshot shows the Query Editor interface with two tabs: 'Console' and 'Results'. The 'Console' tab displays the following ADFL query code:

```

16     placecontext.geo.city,
17     placecontext.geo.countryCode,
18     subwebPageDetails.name AS webPage,
19     TIME_DELTA_IN_MS(PATCH(timestamp, web.WebPageDetails.name 'Call Start', 'seconds'))
20     OVER(PARTITION BY csplatformapac.identification.core.cid
21           ORDER BY timestamp
22           ROWS BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING)
23     AS contact_callcenter_after_seconds
24   FROM aop_demo_luna_telco_website_interactions_dataset
25   WHERE _adlSplitPathForSpout.adlEnvironment.cloudName like 'luna Telco'
26   AND subwebPageDetails.name IN ('Cancel Service', 'Call Start')
27 ) P
28 , aop_demo_luna_telco_call_center_interactions_dataset C
29 , luna_system_profile_dataset_for_crm_telem_v1_1
30 WHERE c.cid = p._adlSplitPathForSpout.cloudRegionId
31 AND c.webPage = 'Cancel Service'

```

The 'Results' tab displays a table with the following data:

| cid | city     | countrycode | latitude | longitude | seconds to contact callcenter | callending | call   |
|-----|----------|-------------|----------|-----------|-------------------------------|------------|--------|
| 1   | Charlton | GB          | 51.59119 | -0.107848 |                               | none       | For... |
| 2   | Charlton | GB          | 51.59119 | -0.107848 |                               | none       | For... |
| 3   | Charlton | GB          | 51.59119 | -0.107848 |                               | none       | For... |

## (Optional) Exercise 8: Create queries in PSQL by using ADF to obtain specific contextual values

In this exercise, you will create the queries, which you created in Exercise 7, in PSQL instead of the Experience Platform UI.

This exercise includes the following tasks:

1. Create a query to obtain the customer actions data on the website before reaching the cancel service page as the third page in a session
2. Create a query to obtain the time lapse data before a visitor calls the call center after visiting the cancel service page
3. Create a query to obtain the outcome of the call center interaction
4. Create a query to obtain the loyalty profile of customers
5. Create a query to obtain the geolocation data of the customers visiting the website

**Task 1:** Create a query to obtain the customer actions data on the website before reaching the cancel service page as the third page in a session

1. Open the **CodeFile\_M4\_Ex8.txt** file and copy the code in the **Module 4 > Exercise 8 > Task 1 > Step 1** section.
2. Open **Command Prompt** and paste the copied code.
3. Press the Enter key to execute the query. The result is displayed, as shown:

| webPage                                | webPage_2                              | webPage_3      | webPage_4      | journeys |
|--|--|----------------|----------------|----------|
| Broadband deals                        | Broadband deals                        | Cancel Service | Cancel Service | 8        |
| Telco Home                             | Telco Home                             | Cancel Service | Cancel Service | 7        |
| Google Pixel XI 128GB Black Smartphone | Google Pixel XI 128GB Black Smartphone | Cancel Service | Cancel Service | 6        |
| Samsung Galaxy S7 32GB Black           | Samsung Galaxy S7 32GB Black           | Cancel Service | Cancel Service | 5        |
| SIM Only                               | SIM Only                               | Cancel Service | Cancel Service | 4        |
| Samsung Galaxy S8                      | Samsung Galaxy S8                      | Cancel Service | Cancel Service | 4        |
| TV & Broadband Deals                   | TV & Broadband Deals                   | Cancel Service | Cancel Service | 4        |
| Luna Telco Sport                       | Luna Telco Sport                       | Cancel Service | Cancel Service | 2        |
| Luna Telco Shop                        | Luna Telco Shop                        | Cancel Service | Cancel Service | 2        |
| (9 rows)                               |  |                |                |          |

Task 2: Create a query to obtain the time lapse data before a visitor calls the call center after visiting the cancel service page

1. Open the **CodeFile\_M4\_Ex8.txt** file and copy the code in the **Module 4 > Exercise 8 > Task 2 > Step 1** section.
  2. Open **Command Prompt** and paste the copied code.
  3. Press the Enter key to execute the query. The result is displayed, as shown:

| event_id                          | webPage        | contact_recallcenter_after_seconds |
|-----------------------------------|----------------|------------------------------------|
| 993318866289679929148847965692117 | Cancel Service |                                    |
| 993318866289679929148847965692117 | Cancel Service | -797                               |
| 993318866289679929148847965692117 | Cancel Service | -797                               |
| 993318866289679929148847965692117 | Cancel Service | -519                               |
| 993318866289679929148847965692117 | Cancel Service | -519                               |
| 993318866289679929148847965692117 | Cancel Service | -67                                |
| 993318866289679929148847965692117 | Cancel Service | -62                                |
| 99752893837616244422322357218955  | Cancel Service |                                    |
| (15 rows)                         |                |                                    |

Task 3: Create a query to obtain the outcome of the call center interaction

1. Open the **CodeFile\_M4\_Ex8.txt** file and copy the code in the **Module 4 > Exercise 8 > Task 3 > Step 1** section.
  2. Open **Command Prompt** and paste the copied code.
  3. Press the Enter key to execute the query. The result is displayed, as shown:

**Task 4:** Create a query to obtain the loyalty profile of customers

1. Open the **CodeFile\_M4\_Ex8.txt** file and copy the code in the **Module 4 > Exercise 8 > Task 4 > Step 1** section.
  2. Open **Command Prompt** and paste the copied code.
  3. Press the Enter key to execute the query. The result is displayed, as shown:

Task 5: Create a query to obtain the geolocation data of the customers visiting the website

1. Open the **CodeFile\_M4\_Ex8.txt** file and copy the code in the **Module 4 > Exercise 8 > Task 5 > Step 1** section.
  2. Open **Command Prompt** and paste the copied code.
  3. Press the Enter key to execute the query. The result is displayed, as shown:

## Exercise 9: Create a query combining online, call center, and loyalty data available in the Experience Platform UI

In the previous exercises, you created a query to analyze visitors who contacted the call center when considering service cancellation. In this exercise, you will create a query and consider all call center interactions including Wi-Fi, promotion, invoice, complaint, and contract to analyze call center interaction.

1. Open the **CodeFile\_M4\_Ex9.txt** file and copy the code in the **Module 4 > Exercise 9 > Step 1** section.
2. Open a text editor. For example, open **Notepad**.
3. Paste the copied code in the Notepad.
4. In the first line of the code, replace the **enter your name** text with your student ID. Do not remove the comment symbols /\* \*/ on either side of your name value.
5. Copy the complete code from the text editor.
6. Open Experience Platform and on the left pane, click **Queries**.
7. Ensure you are in the **Log** tab and click **Create query**. The query editor opens.
8. Paste the query you copied in step 5 and click the Play button to execute the query.
9. Click the **Results** tab. The result is displayed, as shown:

| event | city    | latitude  | longitude | countrycode | callingtype | calltopic | contractCancelled | loyaltystatus | loyaltypoints | email                     |
|-------|---------|-----------|-----------|-------------|-------------|-----------|-------------------|---------------|---------------|---------------------------|
| 1     | Tullich | 574694803 | 33269422  | GB          | positive    | wifi      | no                |               |               | d0is0t0@achieveonginc.com |
| 2     | Tullich | 574694803 | -23269422 | GB          | positive    | wifi      | no                |               |               | frankieart@gmail.com      |
| 3     | Tullich | 574694803 | 33269422  | GB          | positive    | wifi      | no                |               |               | ajolosch@zgawik.com       |

10. Click the **Console** tab to review the execution data. Notice that only the first 100 rows are returned. In this exercise, the total number of rows returned is 995. In the query editor, the maximum number of rows displayed is 100.
11. Click **Cancel** to return to the **Queries** page.

(Optional) Exercise 10: Create a query combining online, call center, and loyalty data available in PSQL

In Exercise 9, you used the Experience Platform query editor to create and execute the query combining online, call center, and loyalty data. In this exercise, you will use PSQL to create and execute the same query.

1. Open the **CodeFile\_M4\_Ex10.txt** file and copy the code in the **Module 4 > Exercise 10 > Step 1** section.
  2. Open a text editor. For example, open **Notepad**.
  3. Paste the copied code in the workspace of the text editor.
  4. In the first line of the code, replace the **enter your name** text with your student ID. Do not remove the comment symbols `/* */` on either side of your name value.
  5. Copy the updated code from the text editor.
  6. Open **Command Prompt** and paste the copied code.
  7. Press the Enter key to execute the query. The result is displayed, as shown:

8. Click x (close) to close the **Command prompt**.

## Exercise 11: Create a dataset from the query result in the Experience Platform UI

Query service takes longer time to generate the results of complex queries. So, it is advisable to generate a dataset from the query results. In this exercise, you will create a new dataset based on the query result from the previous exercises. This new dataset will be used for analysis and visualization in other exercises.

1. On the Experience Platform left navigation pane, navigate to **DATA MANAGEMENT > Queries**. The **Queries** page opens.
2. Click the **Log** tab. The logs are displayed.
3. In the **Search** box, type your student ID that you used in Exercise 9 and 10. For example, **std121**. The queries associated with your student ID is displayed.
4. Click the text in the **CLIENT or CREATED BY** columns. The query details section is displayed on the right, as shown:

| NAME  | CREATED BY | CLIENT                 | STATUS             | DATASET           |
|---|------------|------------------------|--------------------|-------------------|
| select /*^/ e_adisplatform...                 |            | PSQL                   | Returned to client | AEP Demo - Lum... |
| DECLARE psql_js client cursor 0 CURSOR FOR... |            | Adobe Query Service UI | Returned to client | AEP Demo - Lum... |
| select /*^/ e_adisplatform...                 |            | API                    | Dataset created    | AEP Demo - Lum... |
| select /*^/ e_adisplatform...                 |            | PSQL                   | Returned to client | AEP Demo - Lum... |
| DECLARE psql_js_client_cursor_0 CURSOR FOR... |            | Adobe Query Service UI | Returned to client | AEP Demo - Lum... |

5. Click **Output dataset**, as shown. The **Output dataset** dialog opens.

Name: select /\*^/ e\_adisplatform...

**Output dataset**

Delete query

6. In the **Name** box, type <yourstudentID> **Callcenter Interaction Analysis**. For example, **std121 Callcenter Interaction Analysis**.

7. In the **Description** box, type <yourstudentID> **Callcenter Interaction Analysis**. For example, **std121 Callcenter Interaction Analysis**.
8. Click **Run query**. The **Queries** page opens. Notice that the query is listed in the log entries, and the **Submitted** status is displayed, as shown:

| NAME                         | CREATED BY | CLIENT | STATUS             | DATASET       |
|------------------------------|------------|--------|--------------------|---------------|
| select /* / e_adlsplatform.. |            | API    | Submitted          | -             |
| select /* / e_adlsplatform.. |            | PSQL   | Returned to client | AEP Demo - Lu |

9. Refresh the browser every 30 seconds until the **STATUS** column shows as **Dataset created**, as shown:



**Hint:** It may take more than five minutes to create the dataset and display the updated status.

| NAME                         | CREATED BY | CLIENT | STATUS             | DATASET       |
|------------------------------|------------|--------|--------------------|---------------|
| select /* / e_adlsplatform.. |            | API    | Dataset created    | AEP Demo - Lu |
| select /* / e_adlsplatform.. |            | PSQL   | Returned to client | AEP Demo - Lu |

10. On the left navigation pane, navigate to **DATA MANAGEMENT > Datasets**. The **Datasets** page opens.
11. Click the **Browse** tab, if not open already, to browse the existing datasets.
12. In the **Search** box, type your student ID. For example, **std121**. The newly created dataset is displayed.
13. In the **NAME** column, click the dataset link.. The **Dataset activity** page opens with the dataset summary.

| NAME                                | CREATED             | SOURCE | SCHEMA  |
|-------------------------------------|---------------------|--------|---|
| pav_callcenter_interaction_analysis | 10/06/2022, 6:57 AM | File   | Some(Adhoc XDM Schema for dataset pav_callcenter_interaction_an |

14. At the upper-right corner, click **Preview dataset**. The preview of the dataset is displayed in a dialog. Notice that the data in the dataset is the same as the result of the query that you executed in Exercise 9 and 10.

| CITY    | LATITUDE   | COUNTRYCODE | CALLFEELING | CONTRACTCANCELLED |
|---------|------------|-------------|-------------|-------------------|
| Tullich | 57.4694803 | GB          | positive    | no                |
| Tullich | 57.4694803 | GB          | positive    | no                |
| Linton  | 54.0542238 | GB          | none        | no                |
| Linton  | 54.0542238 | GB          | none        | no                |
| Langley | 51.888151  | GB          | none        | no                |
| Langley | 51.888151  | GB          | none        | no                |

15. Click **Close** at the upper-right corner to close the preview dialog.

## Exercise 12: Create a visualization report in Power BI Desktop by using the Experience Platform dataset

In this exercise, you will use Microsoft Power BI Desktop to access the Query service UI in Experience Platform and create reports. You can share these reports with your stakeholders.

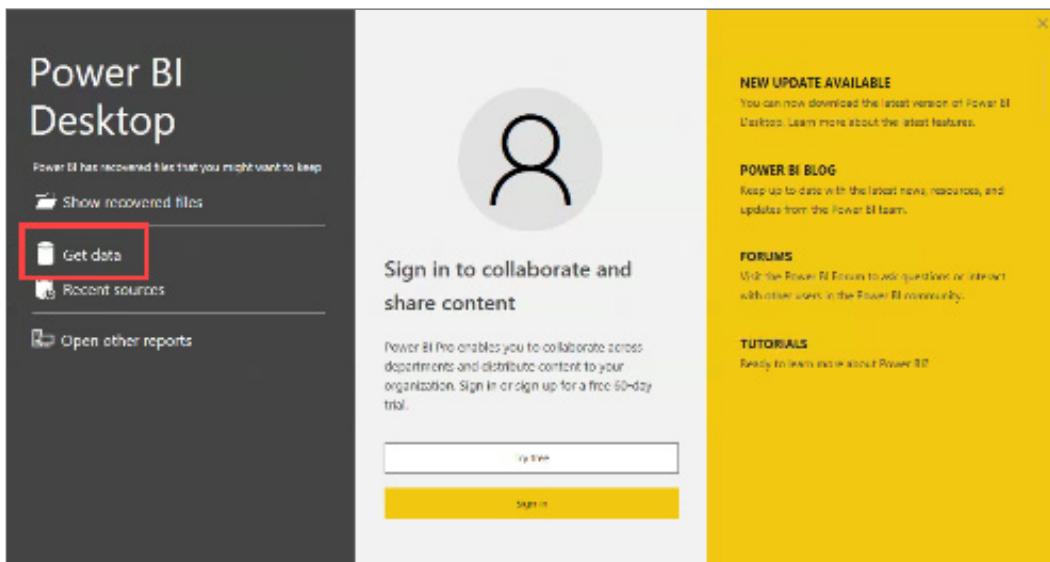
This exercise includes the following tasks:

1. Connect Power BI Desktop to the Query service UI in Experience Platform
2. Create a visualization report in Power BI by using the Experience Platform dataset

### Task 1: Connect Power BI Desktop to the Query service UI in Experience Platform

 **Note:** Ensure that you have installed the **Power BI Desktop** application on your computer with the help of your Instructor.

1. In the Windows search bar, type **Power BI Desktop**. The search result is displayed.
2. Click **Power BI Desktop**. The **Power BI Desktop** application opens.
3. On the left, click **Get data**, as shown. The **Get data** dialog opens.



4. In the search box, type **PostgreSQL**. The search results are displayed.
5. Select **PostgreSQL database** and click **Connect**. The **PostgreSQL database** dialog opens.

6. In the Experience Platform's left navigation pane, navigate to **DATA MANAGEMENT > Queries**.  
The **Queries** page opens.

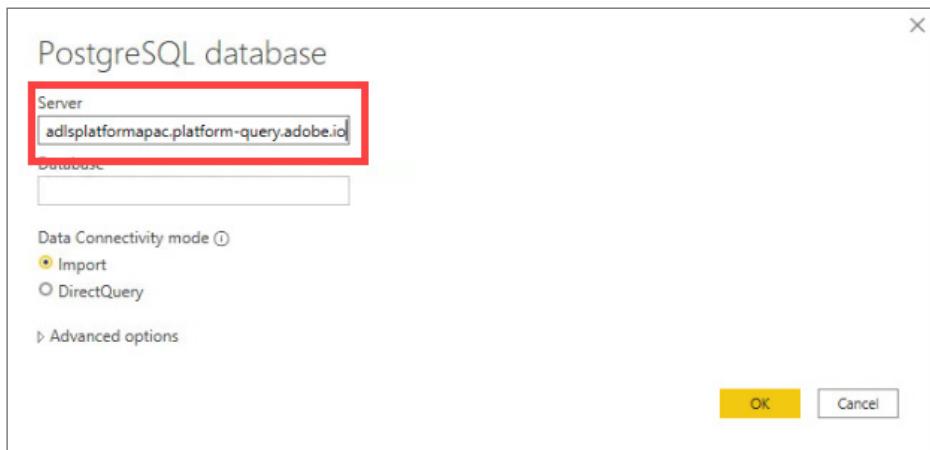


**Note:** Ensure the correct Experience Platform instance is selected.

7. Click the **Credentials** tab. The **PostgreSQL credentials** section is displayed.
8. Click the **Copy to clipboard** icon to copy the **Host** address, as shown:



9. Open **Power BI Desktop** and paste the address in the **Server** box, as shown:



10. Place the cursor at the end of the value in the **Server** box and type **:80**.



**Note:** You must include the port **:80** at the end of the server address because the Query service does not currently use the default PostgreSQL port of 5432.

11. Open Experience Platform and click the **Copy to clipboard** icon for **Database** to copy the database name.
12. Open **Power BI Desktop** and paste the name in the **Database** box.
13. Retain the default selection for the **Data Connectivity mode** section and click **OK**. The **PostgreSQL database** dialog opens.
14. Open Experience Platform and click the **Copy to clipboard** icon for **Username** to copy the user name.
15. Open **Power BI Desktop** and paste the name in the **User name** box.

16. Open Experience Platform and click the **Copy to clipboard** icon for **Password** to copy the password.

17. Open **Power BI Desktop** and paste the name in the **Password** box.

18. Click **Connect**. The **Navigator** dialog opens listing the existing datasets.

## Task 2: Create a visualization report in Power BI Desktop by using the Experience Platform dataset

1. In the **Navigator** dialog, perform the following steps:

a. In the search box, type your student ID. For example, **std121**. The available search results are displayed.

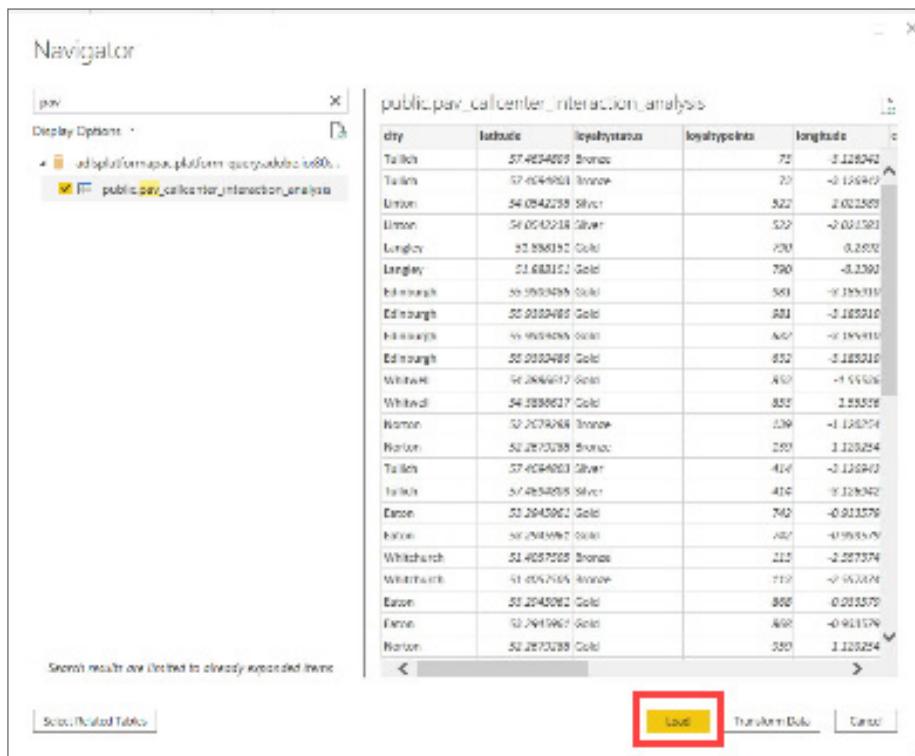
---

 Note: If you cannot find the dataset associated with your student ID, you can select the **public.backup\_callcenter\_interaction\_analysis** available in the list.

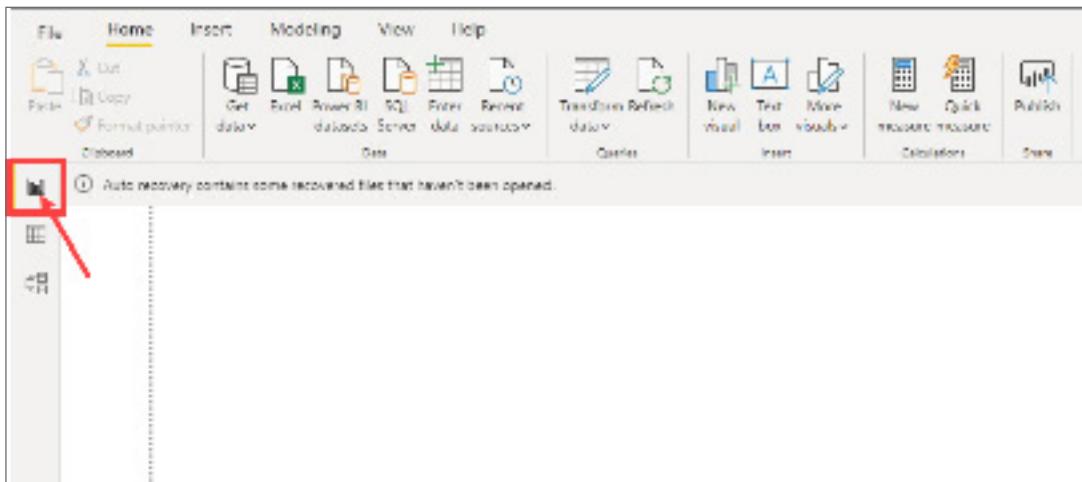
---

b. Select the respective checkbox for your database. For example, **public.std121\_callcenter\_interaction\_analysis**. The dataset preview is displayed on the right.

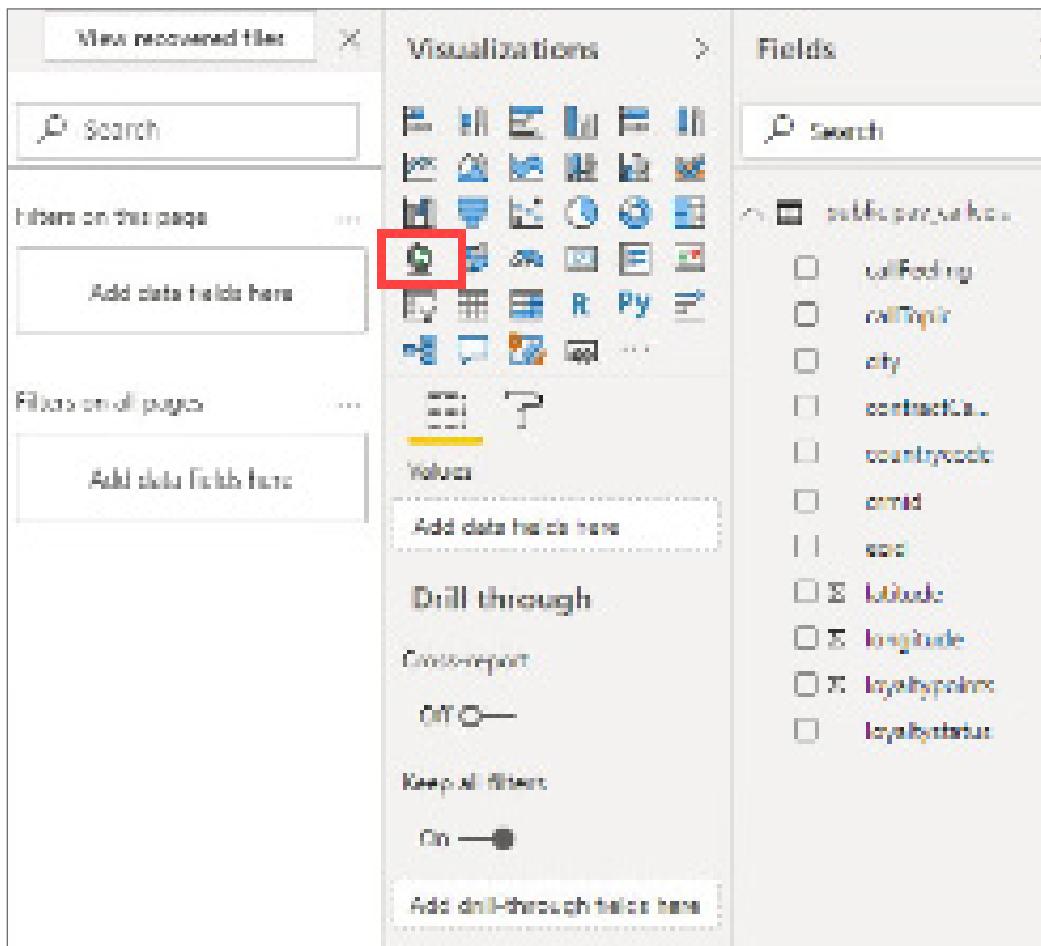
c. Click **Load**, as shown. The **Home** tab opens. It may take a few minutes to load the tab.



2. Ensure that the **Report** icon is selected on the left. The selected dataset fields are listed at the right side in the **Fields** section.

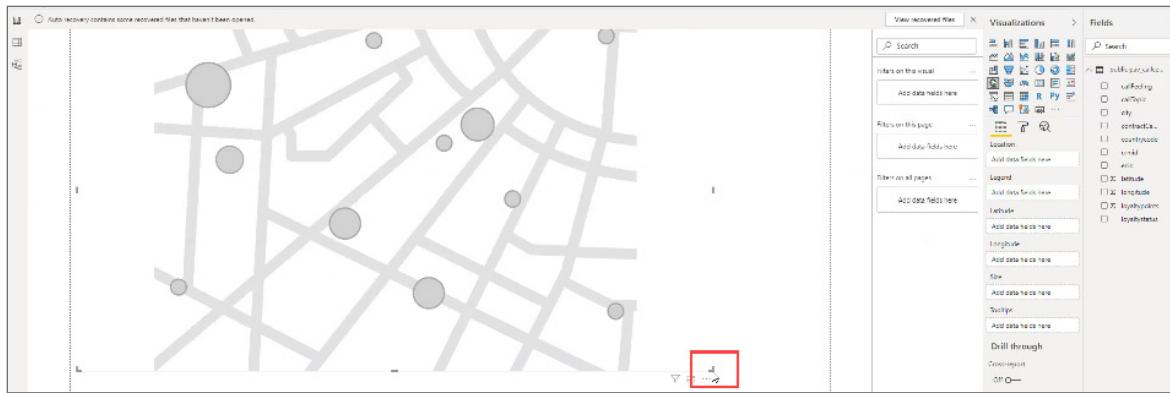


3. In the **Visualizations** section on the right, click the **Map** icon, as shown. The map visualization is added to the reporting canvas.

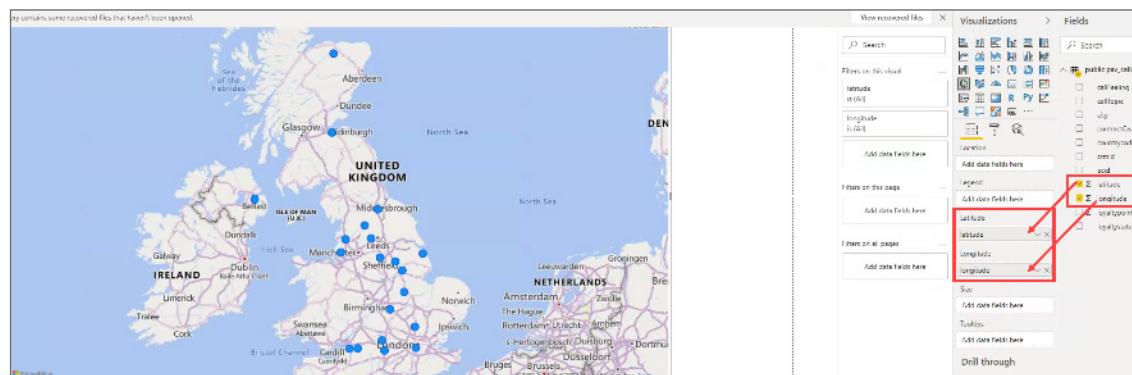


 **Note:** There are tooltips to help you locate each visualization. Point to each icon to display the name of each visualization.

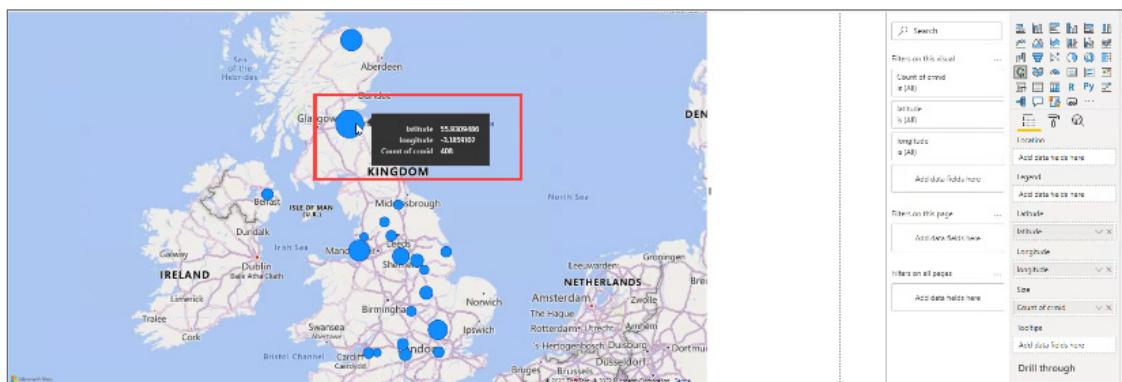
4. Enlarge the map by dragging the edges of the map, as shown:



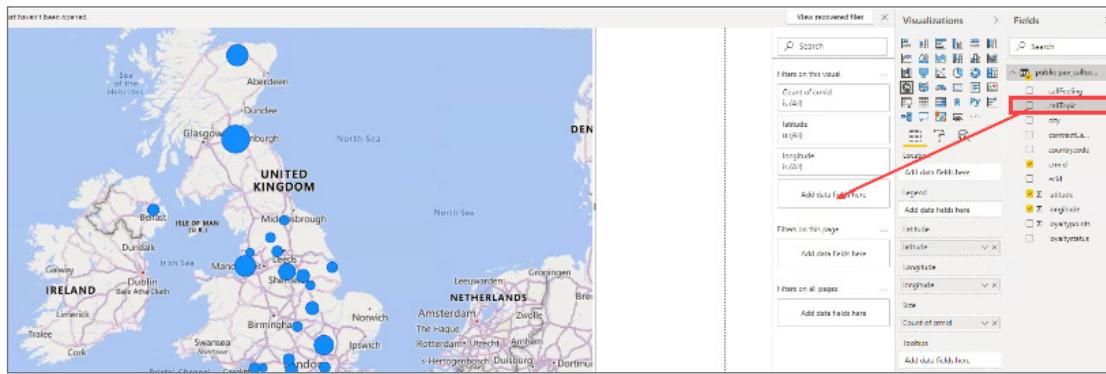
5. From the **Fields** section, drag the **latitude** and **longitude** fields onto their corresponding boxes below the **Visualizations** section, as shown. The map is updated.



6. From the **Fields** section, drag **crmid** onto the **Size** box in the **Visualizations** section. The map is updated. Point the cursor to the blue circles on the map. The **latitude**, **longitude**, and **Count of crmid** are displayed in a tooltip, as shown:



7. From the **Fields** section, drag the **callTopic** field onto the **Add data fields here** box in the **Filters** section, as shown, to perform call topic analysis. The **Filters** section is updated.



**Tip:** You can minimize and maximize the **Filters**, **Fields**, and **Visualization** sections by using the arrows in each section.

8. Select the required **callTopic** checkboxes to visually analyze the data. For example, if **promo** is selected as **callTopic**, the corresponding areas that are under the **promo** callTopic are highlighted in the map.

## Appendix

# Data Science Workspace in Adobe Experience Platform

### Introduction

In the world of digital marketing, data is the key. The amount of data generated every day is increasing exponentially. It is extremely important to analyze the data and make the right predictions of your customer's behavior.

Data Science Workspace in Adobe Experience Platform enables you to generate insights from data. You can use these insights to create personalized recommendations for your customers.

### Objectives

After completing this module, you will be able to:

- Describe Data Science Workspace in Experience Platform
- Explain the process of creating a machine learning model
- Describe the Recipe Builder notebooks in JupyterLab
- Analyze datasets in Experience Platform
- Transform data into an Experience Platform dataset by using the JupyterLab notebook
- Load and analyze the training data
- Create a recipe by using the JupyterLab notebook
- Train and test a machine learning model

## Data Science Workspace: An Overview

---

Data Science Workspace in Experience Platform enables you to use machine learning models to:

- Simplify and accelerate the process of finding insights in data
- Predict customer behavior from the insights
- Create intelligent services to help automate personalized and targeted end-user experiences
- Deliver advanced workflow management, model management, and scalability

# Creating a Machine Learning Model: The Process

---

The process of creating a machine learning model in Data Science Workspace includes:

1. Preparing the data
2. Authoring a recipe
3. Training and evaluating a model
4. Operationalizing the model

## Preparing the data

In the data preparation step, you can:

- Access and explore the data stored in Experience Platform. You can access the common big data and deep learning libraries, such as Spark ML and TensorFlow.
- Ingest your own datasets by using the Experience Data Model (XDM) schema.

Experience Platform enables you to analyze data by using the JupyterLab notebooks. You can:

- Use R, Python, and Scala languages to browse the data catalog on Experience Platform
- Explore data and build models by using the built-in data visualization capability and machine learning libraries
- Use the out-of-the-box templates to quickly understand and visualize the data

## Authoring a recipe

Recipes are machine learning algorithms and logics designed to solve specific business problems.

Experience Platform enables you to author recipes in three ways:

- Use a prebuilt recipe. You can configure the recipe based on your requirement.
- Create a recipe from scratch.
- Upload a recipe created outside Experience Platform. You can also import a recipe from an external repository.

## Training and evaluating a model

A model is an instance of a recipe. You can create, train, and evaluate a machine learning model after authoring a recipe. You can train models by applying different hyper-tuning parameters and evaluate the performance of each model. Data Science Workspace provides a dynamic interface instantly to

visualize and compare model evaluation metrics across multiple test runs. You can identify the top performing training run by analyzing the evaluation metrics.

## Operationalizing the model

When the training run is completed, the top performing training run is identified and scored. Scoring is the process of gathering insights from data by using a trained model. Scoring results are stored in a specified output dataset.

The trained and evaluated model is deployed as an intelligent service through Adobe I/O. The intelligent service uses updated models for predictions. These predictions are added into the Real-time Customer Profile to provide the 360-degree view of the end users.

## Recipe Builder Notebook in JupyterLab

---

The Recipe Builder notebook in JupyterLab provides templates to:

- Perform training and scoring runs inside the notebook
- Edit the training and scoring methods when running experiments on the training and scoring data
- Create a recipe by using the JupyterLab notebook that can be used in Data Science Workspace

## Exercise 1: Analyze datasets in Experience Platform

In this exercise, you will analyze the dataset in Experience Platform that has data streamed from the website. To develop a product recommendation machine learning model, you need to analyze the historic data about the products purchased by the customers on the website.

The table below displays the details of the datasets that you will refer in this exercise:

| Dataset Name  | Dataset Schema                                       | Description  |
|---|--|--|
| Demo System - Event Dataset for Website (Global v1.1) | Demo System - Event Schema for Website (Global v1.1) | Dataset with clickstream data from a website.  |
| AEP Demo - Recommendations Input Dataset              | AEP Demo - Recommendations Input Schema              | Dataset that is used to train the Recommendations machine learning model.                    |
| AEP Demo - Recommendations Output Dataset             | AEP Demo - Recommendations Output Schema             | Dataset obtained after scoring that contains the list of recommended products for each user. |

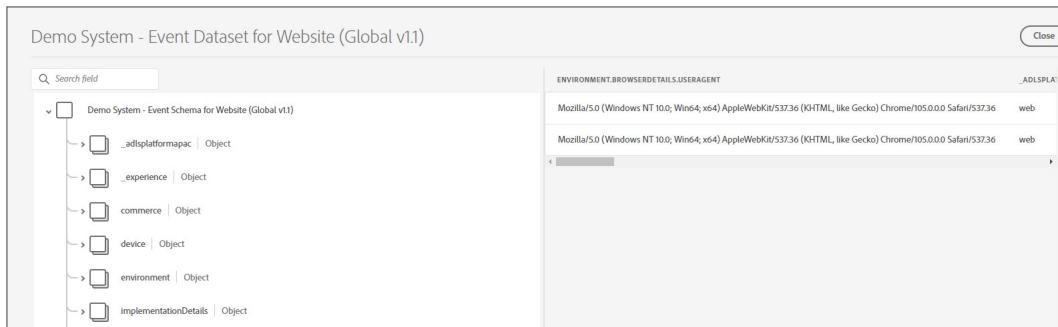
This exercise includes the following tasks:

1. Verify the dataset streamed from a website into Experience Platform
2. Load the training data files into the JupyterLab notebooks to analyze data

### Task 1: Verify the dataset streamed from a website into Experience Platform

1. Log in to <https://platform.adobe.com>. The **Adobe Experience Platform** page opens.
2. On the left pane, navigate to **DATA MANAGEMENT > Datasets**. The **Datasets** page opens.
3. Ensure you are on the **Browse** tab.
4. To review the **Demo System - Event Dataset for Website (Global v1.1)** dataset, perform the following steps:
  - a. In the **Search** box, type **website**. The search results are displayed.
  - b. In the **NAME** column, click the **Demo System - Event Dataset for Website (Global v1.1)** link. The **Demo System - Event Dataset for Website (Global v1.1)** dataset page opens.

- c. At the upper-right corner, click **Preview dataset**. The **Demo System - Event Dataset for Website (Global v1.1)** dialog box opens with the preview of the dataset, as shown:

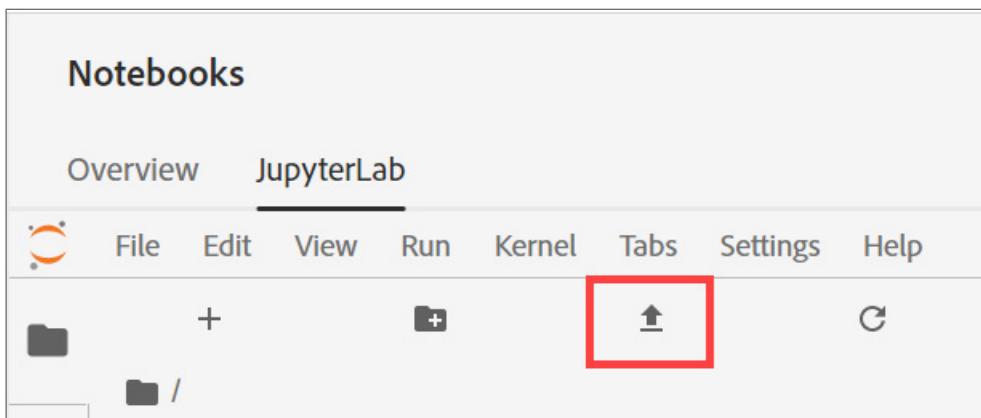


- d. Scroll through the preview content and click **Close** to return to the **Demo System - Event Dataset for Website (Global v1.1)** dataset page.

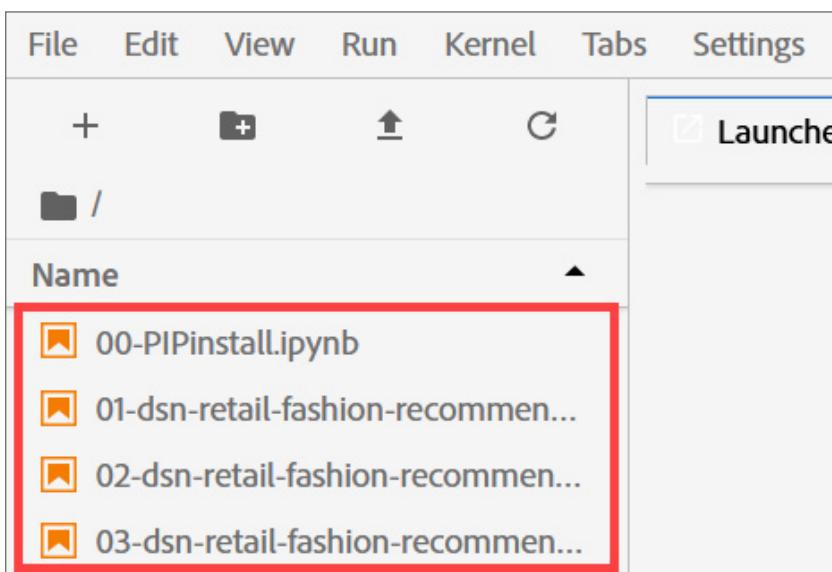
5. To review the **AEP Demo - Recommendations Input** dataset, perform the following steps:
  - a. Navigate to **DATA MANAGEMENT > Datasets**. The **Datasets** page opens on the **Browse** tab.
  - b. In the **Search** box, type **Input**. The search results are displayed.
  - c. In the **NAME** column, click the **AEP Demo - Recommendations Input Dataset** link. The **AEP Demo - Recommendations Input Dataset** page opens.
  - d. At the upper-right corner, click **Preview dataset**. The **AEP Demo - Recommendations Input Dataset** dialog box opens with the preview of the dataset.
  - e. Scroll through the preview content and click **Close** to return to the **AEP Demo - Recommendations Input Dataset** page.
6. To review the **AEP Demo - Recommendations Output Dataset**, perform the following steps:
  - a. Navigate to **DATA MANAGEMENT > Datasets**. The **Datasets** page opens on the **Browse** tab.
  - b. In the **Search** box, type **Output**. The search results are displayed.
  - c. In the **NAME** column, click the **AEP Demo - Recommendations Output Dataset** link. The **AEP Demo - Recommendations Output Dataset** page opens.
  - d. At the upper-right corner, click **Preview dataset**. The **AEP Demo - Recommendations Output Dataset** dialog box opens with the preview of the dataset.
  - e. Scroll through the preview content and click **Close** to return to the **AEP Demo - Recommendations Output Dataset** page.

## Task 2: Load the training data files into the JupyterLab notebooks to analyze data

1. On the left navigation pane, navigate to **DATA SCIENCE > Notebooks**. The **Notebooks** page opens.
2. Click the **JupyterLab** tab. The **Notebooks** page opens. It might take a few minutes for the page to load.
3. On the toolbar, click the **Upload Files** icon, as shown. The dialog box to select and upload the file opens.



4. Navigate to the folder where you saved the exercise files for your region.
5. In the folder, open the JupyterLab notebook files and select all four files:
  - > **00-PIPinstall.ipynb**
  - > **01-dsn-retail-recommendations-feature-transformation.ipynb**
  - > **02-dsn-retail-recommendations-data-exploration.ipynb**
  - > **03-dsn-retail-recommendations-popularity-recipe.ipynb**
6. Click **Open** to upload the files to the **Notebooks** page, as shown:



## Exercise 2: Transform data into an Experience Platform dataset by using the JupyterLab notebook

---

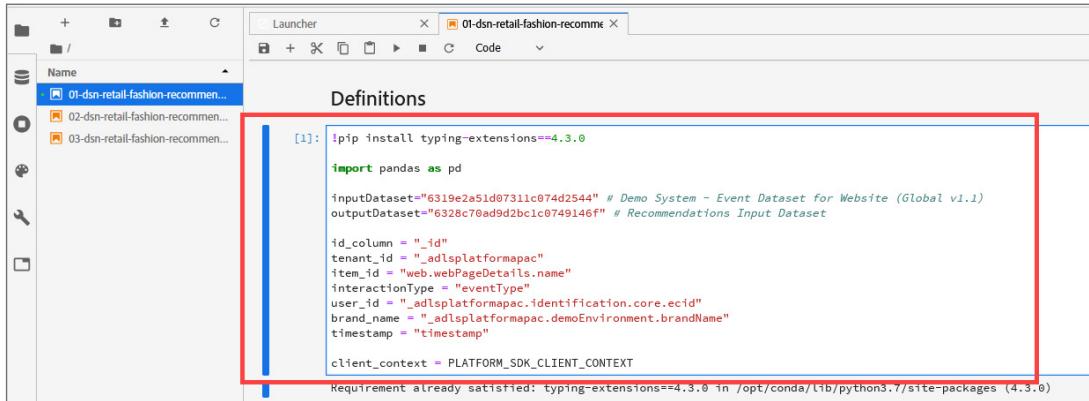
In this exercise, you will define the data, create a new data frame, and transform the data into an Experience Platform dataset.

This exercise includes the following tasks:

1. Define the input and output datasets
2. Read the website clickstream data and load the data
3. Filter out empty values
4. Transform the data by splitting products into individual records
5. View a sample of the split data
6. Prepare the data by adding additional columns required for your model
7. Verify a sample of the prepared data
8. Ingest the data to a dataset in Experience Platform
9. Verify the ingested data in Experience Platform

## Task 1: Define the input and output datasets

1. Double-click the **00-PIPinstall.ipynb** notebook. A new tab opens on the right.
2. Click the **Run the selected cells and advance** button (the play button). The code is executed.
3. Double-click the **01-dsn-retail-recommendations-feature-transformation.ipynb** notebook. A new tab opens on the right.
4. Click anywhere in the code box below **Definitions**. The box is highlighted with a blue border, as shown:



The screenshot shows a Jupyter Notebook interface with a sidebar containing file navigation. The main area has a tab titled "01-dsn-retail-fashion-recomm..." and a sub-tab "Code". Below the tabs, the word "Definitions" is centered. A code cell is displayed with a blue vertical bar on the left and a red rectangular border around the code content. The code itself is as follows:

```
[1]: !pip install typing-extensions==4.3.0
import pandas as pd

inputDataset="6319e2a51d07311c074d2544" # Demo System - Event Dataset for Website (Global v1.1)
outputDataset="6328c79ad9d2bc1c0749146f" # Recommendations Input Dataset

id_column = "_id"
tenant_id = ".adlsplatformapac"
item_id = "web.webPageDetails.name"
interactionType = "eventType"
user_id = ".adlsplatformapac.identification.core.ecid"
brand_name = ".adlsplatformapac.demoEnvironment.brandName"
timestamp = "timestamp"

client_context = PLATFORM_SDK_CLIENT_CONTEXT
```

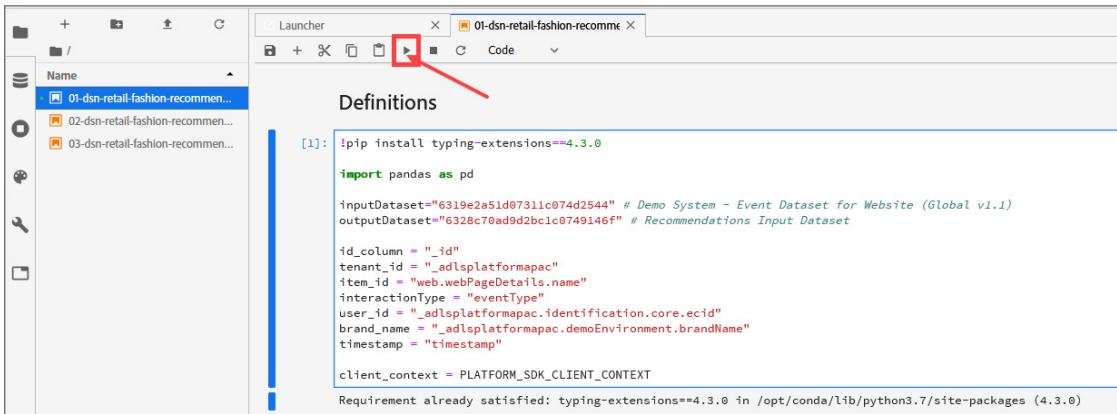
Requirement already satisfied: typing-extensions==4.3.0 in /opt/conda/lib/python3.7/site-packages (4.3.0)

---

 **Note:** The tenant ID and item ID values in code will be different for each region.

---

5. Click the **Run the selected cells and advance** button (the play button), as shown. The code is executed. Alternatively, you can execute the code by pressing the Shift+Enter keys. The execution of this cell might take a few minutes.



```
[1]: !pip install typing-extensions==4.3.0
import pandas as pd

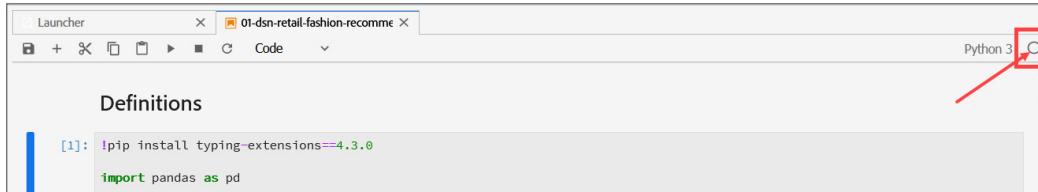
inputDataset="6319e2a51d07311c074d2544" # Demo System - Event Dataset for Website (Global v1.1)
outputDataset="6328c70ad9d2bc1c0749146f" # Recommendations Input Dataset

id_column = "_id"
tenant_id = "_adlsplatformapac"
item_id = "web.webPageDetails.name"
interactionType = "eventType"
user_id = "_adlsplatformapac.identification.core.ecid"
brand_name = "_adlsplatformapac.demoEnvironment.brandName"
timestamp = "timestamp"

client_context = PLATFORM_SDK_CLIENT_CONTEXT
Requirement already satisfied: typing-extensions==4.3.0 in /opt/conda/lib/python3.7/site-packages (4.3.0)
```

After completion, the execution complete indications are displayed. Following are the indicators for cell execution status:

- › A full circle beside the Python code at the upper-right corner indicates that the execution is still running.
- › An empty circle beside the Python code at the upper-right corner, as shown, indicates that the execution is complete.



```
[1]: !pip install typing-extensions==4.3.0
import pandas as pd
```

- › A star inside the square brackets beside cell indicates the execution is still running.
- › A sequence number displayed inside the square brackets beside cell indicates the completed execution order for the cell.

---

**Caution:** Do not continue the exercise until an indicator shows that the execution is completed. If you do not wait for the execution to finish, you will encounter errors in the next steps. This is applicable to the execution of all cells in any JupyterLab notebook.

---

## Task 2: Read the website clickstream data and load the data

1. On the **01-dsn-retail-recommendations-feature-transformation.ipynb** tab, scroll down to the **Load the event data** section.
2. click the code in the box and click the **Run the selected cells and advance** button to execute the code. The execution might take a few minutes. You need to wait until the execution is completed. The result is displayed, as shown:

Load the event data

```
from platform_sdk.dataset_reader import DatasetReader
dataset_reader = DatasetReader(client_context, inputDataset)
df = dataset_reader.limit(50000).read()

INFO:PlatformSDKPython:dataset_reader: seconds taken to get dataset details from catalog and make PQS connection: 2.94
INFO:PlatformSDKPython:dataset_id: 6319e2a51d0731c074d2544, limit: 50000
INFO:PlatformSDKPython:dataset_reader: seconds taken to execute query: 22.32
INFO:PlatformSDKPython:dataset_reader: 50000 rows read. 327.3 MB memory used for this process
/opt/conda/lib/python3.7/site-packages/platform_sdk/dataset_reader.py:140: FutureWarning: The pandas.np module is deprecated and will be removed from pandas in a future version. Import numpy directly instead
    df = df.replace({pd.np.nan: None})
INFO:PlatformSDKPython:dataset_reader: seconds taken to format data of dataframe: 2.13
```

3. Scroll down to view the **View a sample of the data in the dataframe** section. The code box for the section is displayed.
4. Click in the code box and click the **Run the selected cells and advance** icon. The code is executed and the result displays the data sample in the data frame, as shown:

### View a sample of the data in the dataframe

```
[]: # Examine the page names in the dataframe.
print(df["web.webPageDetails.name"])

0           men : Montana Wind Jacket
1                           product7
2           men : Montana Wind Jacket
3                           product19
4      equipment : Sprite Yoga Straps
...
49995        Under Armour Boys
49996            My cart
49997   Fashion collection
49998            My cart
49999       Boys Belt CHAOREN
Name: web.webPageDetails.name, Length: 500000, dtype: object
```

### Task 3: Filter out empty values

1. Scroll down to the **Filtering** section, as shown:

### Filtering

```
] : # drop nulls
df = df.dropna(subset=[id_column, user_id, item_id, interactionType, brand_name])

# only focus on one brand
df = df[df[brand_name] == "Retail fashion"]
```

2. Click in the code box and click the **Run the selected cells and advance** icon. The code is executed. However, you cannot view the results in the notebook.

### Task 4: Transform the data by splitting products into individual records

In this task, you will split the items in the item\_id field into individual records and save the result to a new data frame called df2.

1. Scroll down to the **Split items into individual records** section.
2. Click the code box and click the **Run the selected cells and advance** icon. The code is executed. The execution of the code might take a few seconds. You will view the result of this task by executing the next task.

## Task 5: View a sample of the split data

In this task, you will view a sample of the split data in the new data frame. You will verify that the splitting of the individual products in the item\_id field into separate records was successful.

1. Scroll down to the **Sample the split data** section. A code box is displayed, as shown:

### Sample the split data

```
[7]: df2.head(20)
```

2. Select the code and click the **Run the selected cells and advance** icon. The code is executed. Notice that the sample of data in the df2 data frame is displayed, as shown:

| 54 | Retail fashion | Luma | 4155383507555031138370242612871111464  |    |  |
|----|----------------|------|--|----|--|
| 55 | Retail fashion | Luma | 54413560244503224386185874055677852427 |    |  |
| 56 | Retail fashion | Luma | 38877371060304663546507577553457313212 | ZN |  |
| 57 | Retail fashion | Luma | 27663236384713055170420787334782167616 |    |  |
| 58 | Retail fashion | Luma | 35545165671518863334715768704735772810 |    |  |
| 59 | Retail fashion | Luma | 13874277155640403136871547652446113764 |    |  |

## Task 6: Prepare the data by adding additional columns required for your model

1. Scroll down to the **Data Prep for saving back to platform** section.
2. Click in the code box and click the **Run the selected cells and advance** icon. The code is executed. You will view the result of this task by executing the next task.

## Task 7: Verify a sample of the prepared data

1. Scroll down to the **Sample the prepared data** section. The code box is displayed, as shown:

### Sample the prepared data

```
df2.head()
```

2. Select the code and click the **Run the selected cells and advance** icon. The code is executed and the results are displayed, as shown:

### Sample the prepared data

```
df2.head()
```

|    | _adlsplatformapac.demoEnvironment.brandName | _adlsplatformapac.demoEnvironment.brandIndustry | _adlsplatformapac.identification.core.ecid |
|----|---|---|--|
| 54 | Retail fashion                              | Luma  | 4155383507555031138370242612871111464      |
| 55 | Retail fashion                              | Luma  | 54413560244503224386185874055677852427     |
| 56 | Retail fashion                              | Luma  | 388773710603046635465075755345731321       |
| 57 | Retail fashion                              | Luma  | 27663236384713055170420787334782167616     |
| 58 | Retail fashion                              | Luma  | 35545165671518863334715768704735772810     |

## Task 8: Ingest the data to a dataset in Experience Platform

1. Scroll down to the **Write the prepared data to a dataset in platform** section.
2. Click in the code box and click the **Run the selected cells and advance** icon. The code is executed in a few seconds, as shown:

### Write the prepared data to a dataset in platform

```
df2['timestamp'] = pd.to_datetime(df2['timestamp']).apply(lambda x: x.isoformat())

from platform_sdk.models import Dataset
from platform_sdk.dataset_writer import DatasetWriter
dataset = Dataset(PLATFORM_SDK_CLIENT_CONTEXT).get_by_id(dataset_id=outputDataset)
dataset_writer = DatasetWriter(PLATFORM_SDK_CLIENT_CONTEXT, dataset)
write_tracker = dataset_writer.write(df2, file_format='json')

INFO:azure.core.pipeline.policies.http_logging_policy:Request URL: 'https://login.microsoftonline.com:443/fa7b1b5a-7b34-4387-94ae-d2c178dece1/v2.0/.well-known/openid-configuration'
Request method: 'GET'
Request headers:
```

A new batch of data is created and ingested into the **AEP Demo - Recommendations Input Dataset** in Experience Platform.

## Task 9: Verify the ingested data in Experience Platform

In this task, you will verify the status of the batches being ingested into the **AEP Demo - Recommendations Input** dataset in the Experience Platform UI.

1. Navigate to **DATA MANAGEMENT > Datasets**. The **Datasets** page opens.
2. Ensure you are on the **Browse** tab.
3. In the **Search** box, type **Input**. The **AEP Demo - Recommendations Input Dataset** is displayed in the search list.
4. Click the **AEP Demo - Recommendations Input Dataset** link. The **AEP Demo - Recommendations Input Dataset** page opens.

5. Scroll down and notice the date and time in the **INGESTED** column, as shown. The date and time indicate that the data was ingested just now. The date and time may vary depending on the time you executed the ingestion code.

The screenshot shows the Adobe Experience Platform Data Studio interface. At the top, it says "Datasets > AEP Demo - Recommendations Input Dataset". Below that, there are two tabs: "Dataset activity" (which is selected) and "Data governance". Under "Dataset activity", there is a timeline from 1 to 7, with a blue bar indicating activity between week 6 and week 7, specifically on October 7, 2022. The main area displays a table of dataset activity:

| BATCH ID                   | DATAFLOW RUN ID | INGESTED             | RECORDS INGESTED | RECO |
|----------------------------|-----------------|----------------------|------------------|------|
| 01GES2VSZBENNA5GHGTZVQ1FWM | -               | 10/07/2022, 4:51 PM  | 0                | 0    |
| 01GES293VAQJNHXX8RV17Z3C05 | -               | 10/07/2022, 4:41 PM  | 0                | 0    |
| 01GES1360SX79Y78EVY21H28T1 | -               | 10/07/2022, 4:20 PM  | 12,976           | 0    |
| 01GERCHS2VP5YCK5ZY7GMEYN4  | -               | 10/07/2022, 10:21 AM | 12,976           | 0    |

## Exercise 3: Load and analyze the training data

In this exercise, you will analyze the training data to generate insights by using the JupyterLab notebook.

1. On the left pane, navigate to **DATA SCIENCE > Notebooks**. The **Notebooks** page opens.
2. Click the **JupyterLab** tab. The notebooks that you uploaded in Exercise 1 are displayed.
3. Double-click the **02-dsn-retail-recommendations-data-exploration.ipynb** notebook. The notebook opens on a separate tab on the right.
4. In the **Load Libraries** section, click within the code box.
5. Click the **Run the selected cells and advance** icon. The code is executed and the library is loaded, as shown:

**Load Libraries**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import model_selection, linear_model, tree
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.neighbors import KNeighborsRegressor
import warnings; warnings.simplefilter('ignore')
%matplotlib inline
sns.set()
seed = 1234
inputDataset="6328c70ad9d2bc1c0749146f" # Recommendations Input Dataset
```

INFO:matplotlib.font\_manager:generated new fontManager

6. Scroll down to the **Load Data** section and click in the code box.

7. Click the **Run the selected cells and advance** icon. The code is executed and the data from the **AEP Demo - Recommendations Input** dataset is loaded, as shown:

## Load Data

```
from platform_sdk.dataset_reader import DatasetReader
from datetime import date
dataset_reader = DatasetReader(PLATFORM_SDK_CLIENT_CONTEXT, dataset_id=inputDataset)
df = dataset_reader.limit(500).read()

INFO:PlatformSDKPython:dataset_reader: seconds taken to get dataset details from catalog and make PQS connection: 2.85
INFO:PlatformSDKPython:dataset_id: 6328c70ad9d2bc1c0749146f, limit: 500
INFO:PlatformSDKPython:dataset_reader: seconds taken to execute query: 8.82
INFO:PlatformSDKPython:dataset_reader: 500 rows read. 133.17 MB memory used for this process
INFO:PlatformSDKPython:dataset_reader: seconds taken to format data of dataframe: 0.01
```

8. Scroll down to the **Dataset Dimensions** section under **Summarize the Data** and click in the code box.
9. Click the **Run the selected cells and advance** icon. The code is executed and the result displays the number of rows and columns of data, as shown. You can view the sample data in the next step.

## Summarize the Data

### Dataset Dimensions

```
print("You'll see the number of lines and the number of columns:", df.shape)

You'll see the number of lines and the number of columns: (500, 6)
```

10. Scroll down to the **Peeking at the Data** section and click in the code box.
11. Click the **Run the selected cells and advance** icon. The code is executed and the result is displayed, as shown:

## Peeking at the Data

```
df.head(20)
```

|   | _adlsplatformapac.brandName | _adlsplatformapac.interactionType | _adlsplatformapac.itemId | _adlsplat                 |
|---|-----------------------------|-----------------------------------|--------------------------|---------------------------|
| 0 | Retail fashion              | commerce.productViews             | Wide Brim Fedora         | 280460818712410263870367  |
| 1 | Retail fashion              | commerce.productViews             | SOLY HUX Womens          | 5173705000664664236130681 |
| 2 | Retail fashion              | commerce.productViews             | Columbia Boys Glennaker  | 58337775246845080208318   |
| 3 | Retail fashion              | commerce.productViews             | Free to Live             | 88866737400404128053075!  |

12. Scroll down to the **Assign variable names to fields** section and click in the code box.

13. Click the **Run the selected cells and advance** icon. The code is executed. You will not be able to view a result for this execution.
14. Scroll down to the **Statistical summary** section and click in the code box.
15. Click the **Run the selected cells and advance** icon. The code is executed and the result displays the statistical summary of the data, as shown:

| Statistical Summary |                             |                                   |                          |
|---------------------|-----------------------------|-----------------------------------|--------------------------|
| df.describe()       |                             |                                   |                          |
|                     | _adlsplatformapac.brandName | _adlsplatformapac.interactionType | _adlsplatformapac.itemId |
| count               | 500                         | 500                               | 500                      |
| unique              | 1                           | 1                                 | 142                      |
| top                 | Retail fashion              | commerce.productViews             | 420851372244757038115373 |
| freq                | 500                         | 500                               | 41                       |

16. Scroll down to the **Grouping Items by Item ID** section and click in the code box.
17. Click the **Run the selected cells and advance** icon. The code is executed and the result is displayed, as shown:

| Grouping Items by Item ID                                |       |
|--|-------|
| df[[user_id, item_id]].groupby([item_id]).agg(['count']) |       |
| _adlsplatformapac.userId                                 | count |
| _adlsplatformapac.itemId                                 |       |
|  | 41    |
| 2 Pieces Zipper  | 6     |
| 4 Pieces Kids  | 2     |
| 4 Pieces Toddler   | 2     |
| 6 Pieces Zipper  | 2     |
| ...  | ...   |

## Exercise 4: Create a recipe by using the JupyterLab notebook

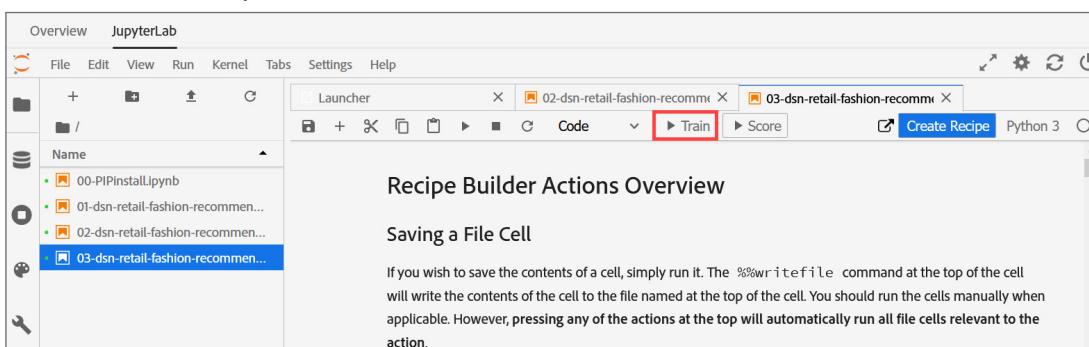
In this exercise, you will train, score, and create a recipe by using the JupyterLab notebook.

This exercise includes the following tasks:

1. Train a recipe in the JupyterLab notebook
2. Score a recipe in the JupyterLab notebook
3. Create a recipe by using the JupyterLab notebook

### Task 1: Train a recipe in the JupyterLab notebook

1. On the left pane, navigate to **DATA SCIENCE > Notebooks**. The **Notebooks** page opens.
2. Click the **JupyterLab** tab. The notebooks that you uploaded in Exercise 1 is displayed.
3. Double-click the **03-dsn-retail-recommendations-popularity-recipe.ipynb** notebook. The notebook opens on a separate tab on the right.
4. Click **Train**, as shown, to perform the training run of the code. The training run starts and takes a few minutes to complete the execution.



The execution of the training data generates evaluation metrics to gauge the machine learning model performance. The result displays a log of commands and outputs from the training script and information of the training run, as shown:

The screenshot shows the JupyterLab interface with a terminal window open. The terminal window title is '03-dsn-retail-fashion-recomm...'. The command entered is 'bash -e run\_action.sh recipe -W qewYF training'. The terminal output shows the following log messages:

```
RUNNING TRAINING...
Ignore the exception for python runtime cannot import name 'BlockBlobService' from 'azure.storage.blob' (/opt/conda/envs/usermlruntime/python/lib/python3.7/site-packages/azure/storage/blob/_init__.py)
2022-10-07 12:23:09,338 INFO trainingInitiator.py: Trainer initiated
INFO:trainingInitiator.py:Trainer initiated
2022-10-07 12:23:09,343 INFO trainingInitiator.py: Evaluator initiated
INFO:trainingInitiator.py:Evaluator initiated
2022-10-07 12:23:09,343 INFO main.py: Training starts, testing:True, conf:/tmp/tmp.00EYqMExtr/token.conf
INFO:main.py:Training starts, testing:True, conf:/tmp/tmp.00EYqMExtr/token.conf
2022-10-07 12:23:09,345 INFO trainingInitiator.py: Training class is not of type Tensorflow
INFO:trainingInitiator.py:Training class is not of type Tensorflow
2022-10-07 12:23:09,345 INFO trainingInitiator.py: Python Job
INFO:trainingInitiator.py:Python Job
2022-10-07 12:23:09,345 INFO trainingInitiator.py: Load the dataframe
```

5. Scroll down to view the **TRAINING SUCCESSFUL!** message at the end of the result, as shown:

The screenshot shows the JupyterLab interface with the following details:

- File Explorer:** On the left, it lists notebooks: "00-PIPInstall.ipynb", "01-dsn-retail-fashion-recomm...", "02-dsn-retail-fashion-recomm...", and "03-dsn-retail-fashion-recomm...". The "03-dsn-retail-fashion-recomm..." notebook is currently selected.
- Launcher:** At the top center, there are tabs for "Launcher", "02-dsn-retail-fashion-recomm...", and "03-dsn-retail-fashion-recomm...".
- Code Editor:** The main area displays Python training logs:

```
INFO:trainingInitiator.py:Python Training completed  
2022-10-07 12:23:27,734 INFO trainingInitiator.py: Evaluation will be on the test data  
INFO:trainingInitiator.py:Evaluation will be on the test data  
2022-10-07 12:23:27,734 INFO trainingInitiator.py: Evaluate config is set to true  
INFO:trainingInitiator.py:Evaluate config is set to true  
2022-10-07 12:23:27,737 INFO evaluateInitiator.py: Starting evaluation  
INFO:evaluateInitiator.py:Starting evaluation  
Evaluation evaluate triggered  
[{'name': 'Recall', 'value': 0.18907305095150398, 'valueType': 'double'}, {'name': 'Precision', 'value': 0.061513965623081646, 'valueType': 'double'}]  
2022-10-07 12:23:28,335 INFO evaluateInitiator.py: Evaluation completed  
INFO:evaluateInitiator.py:Evaluation completed
```
- Message:** A red box highlights the text "TRAINING SUCCESSFUL!" at the bottom of the log area.

The following cells are executed during the training run:

- › Requirements file
  - › Configuration files
  - › Training Data Loader file
  - › Scoring Data Loader file
  - › Pipeline file
  - › Evaluator file
  - › Data Saver file

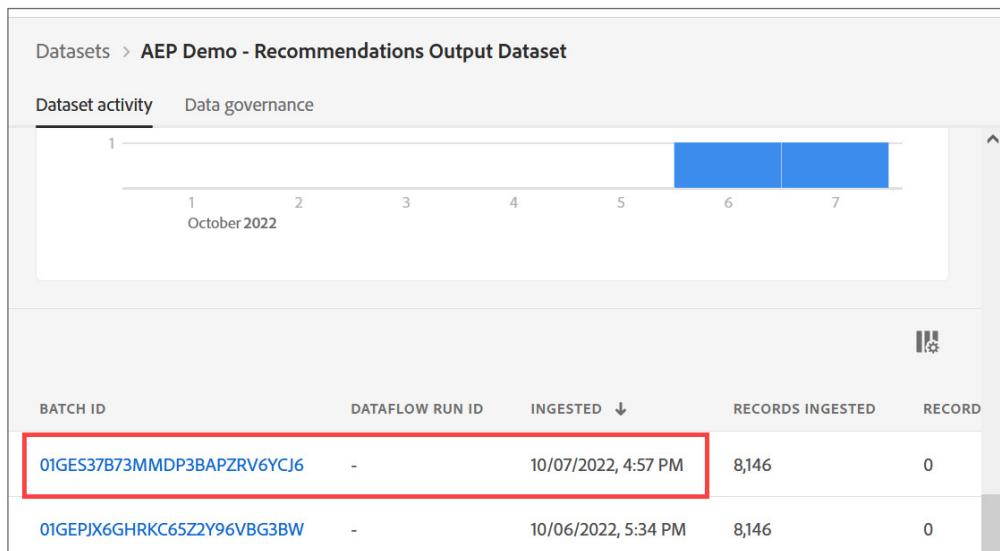
## Task 2: Score the recipe in the JupyterLab notebook

1. Click **Score** in the **03-dsn-retail-recommendations-popularity-recipe.ipynb** tab. The scoring run starts. It takes a few minutes to complete the execution.
2. Scroll down to view the **SCORING SUCCESSFUL!** message. The scoring run output with recommendations is stored in the **AEP Demo - Recommendations Output** dataset in Experience Platform.

The cells executed during the scoring run are:

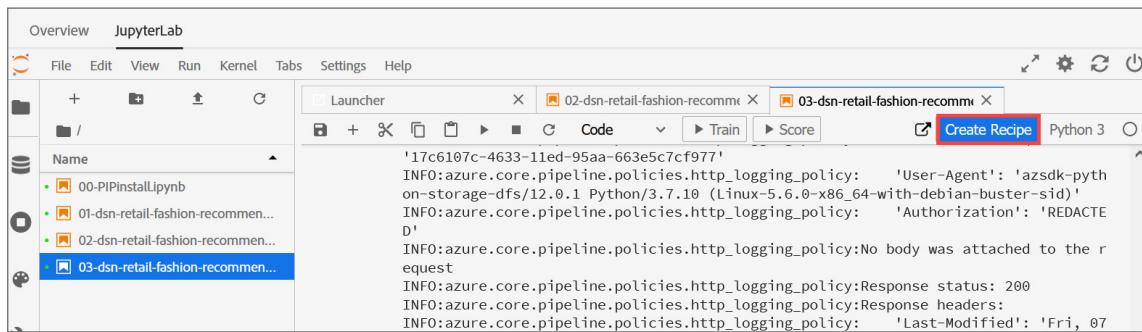
- › Requirements file
- › Configuration files - Scoring
- › Scoring Data Loader file
- › Pipeline file
- › Evaluator file

3. To verify the scored output in Experience Platform, perform the following steps:
  - a. Navigate to **DATA MANAGEMENT > Datasets**. The **Datasets** page is displayed.
  - b. Click the **Browse** tab, if not open already. The existing datasets are displayed.
  - c. In the **Search** box, type **AEP Demo - Recommendations Output**. The dataset is listed in the search results.
  - d. Click **AEP Demo - Recommendations Output**. The **AEP Demo - Recommendations Output** dataset page opens.
  - e. Scroll down and notice that a successful batch was ingested a few minutes ago, as shown:

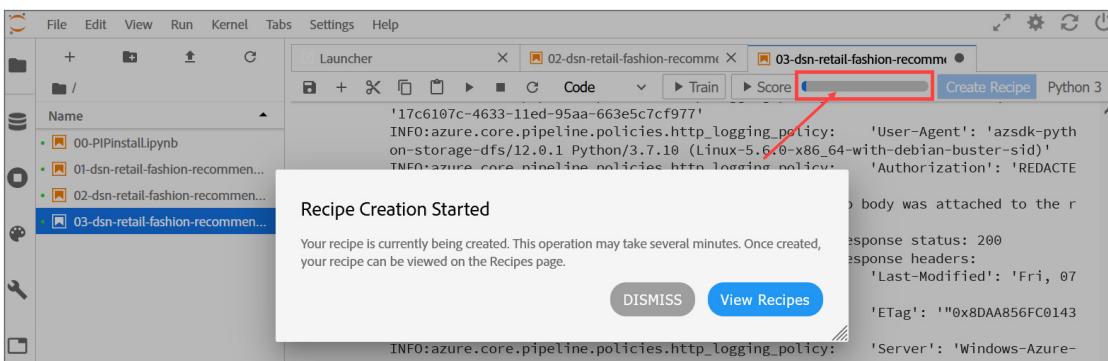


### Task 3: Create a recipe by using the JupyterLab notebook

1. On the left navigation pane, navigate to **Notebooks > JupyterLab**. The available notebooks are listed.
2. Double-click the **03-dsn-recommendations-popularity-recipe.ipynb** notebook. The notebook opens on a separate tab.
3. Click **Create Recipe**, as shown, to start the process. The **Enter Recipe Name** dialog box opens.

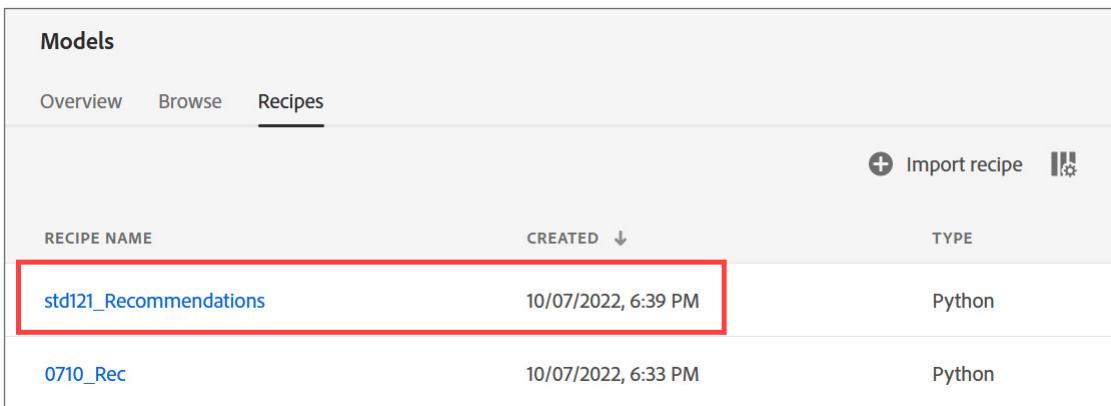


4. Type a name for the recipe. For example, **std121\_Recommendations**.
5. Click **OK**. The **Recipe Creation Started** dialog box opens indicating that the recipe creation has started.
6. This process may take a few minutes. You must wait until the process completes. You can view the progress of the recipe creation process at the upper-right corner, as shown:



After a few minutes, the recipe creation is completed.

7. After the recipe creation is completed, in the **Recipe Creation Started** dialog box, click **View Recipes**. The **Models** page opens on the **Recipes** tab. The newly created recipe is displayed in the list, as shown:



| RECIPE NAME            | CREATED             | TYPE   |
|------------------------|---------------------|--------|
| std121_Recommendations | 10/07/2022, 6:39 PM | Python |
| 0710_Rec               | 10/07/2022, 6:33 PM | Python |

Stay on this page for the next exercise.

## Exercise 5: Train and test a machine learning model

---

You have prepared your data, authored your recipe, and packaged it to test.

This exercise includes the following tasks:

1. Create a machine learning model in Data Science Workspace
2. Train the machine learning model
3. Score the machine learning model

### Task 1: Create a machine learning model in Data Science Workspace

1. On the **Recipes** tab, press the Ctrl+F keys, and type your studentID to find the recipe that you created in the previous exercise. For example, **std121**. The search list is displayed.
2. Click the link in the **RECIPE NAME** column of your recipe. For example, **std121\_Recommendations**. The recipe page opens.
3. At the upper-right corner, click **Create model**. The **Create model** page opens.

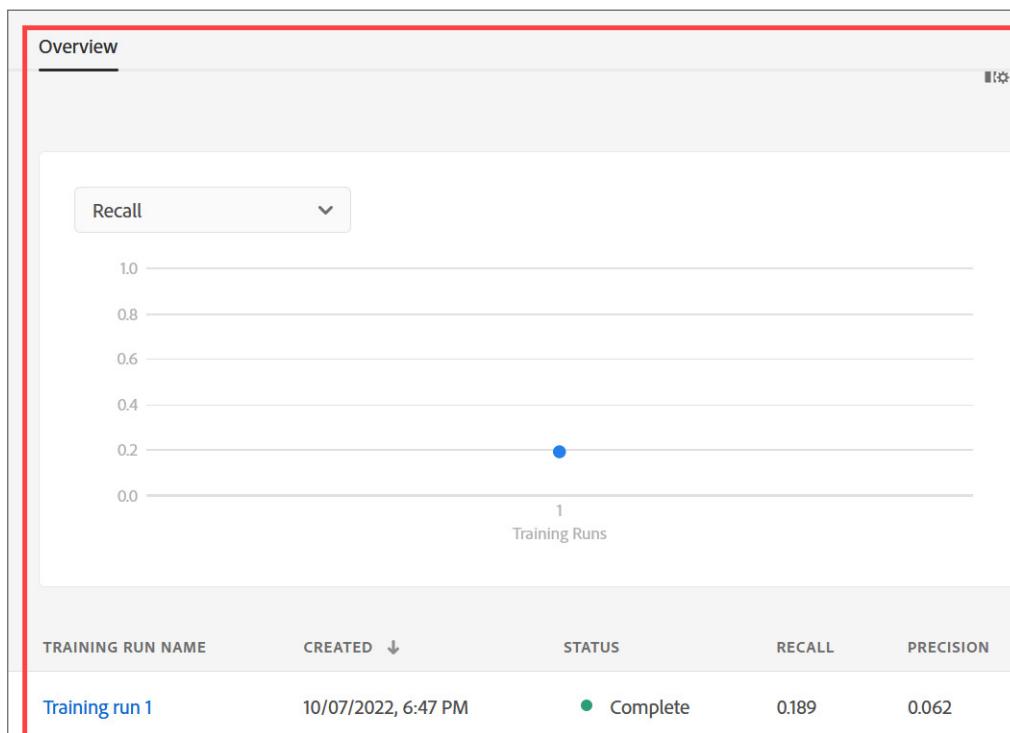
### Task 2: Train the machine learning model

To train the model, you will define an input dataset. In this task, you will use the generated data from the previous exercise as it contains the information about the product purchase data.

1. Select **AEP Demo - Recommendations Input** from the list. The **AEP Demo - Recommendations Input** pane is displayed on the right.
2. Click **Next**. The page updates to display the **Configure** section.
3. To provide a name for the model and set up other parameters, perform the following steps:
  - a. In **Model name** field, type a name for the model. For example, **std121\_Recommendations**.
  - b. (Optional) On the **Training parameters** tab, scroll down to view the **num\_recommendations** key and value.
  - c. (Optional) Double-click the **VALUE** column for **num\_recommendations** and type **3**.
  - d. (Optional) Double-click the **VALUE** column for **sampling\_fraction** and change to **0.8**.

Click **Finish**. After a few seconds, the newly created model opens. The page displays **Training run 1** with the **Pending** status. The process to finish the training run can take more than five minutes. After the training run is completed, the status first updates to **Running** and finally to **Complete**.

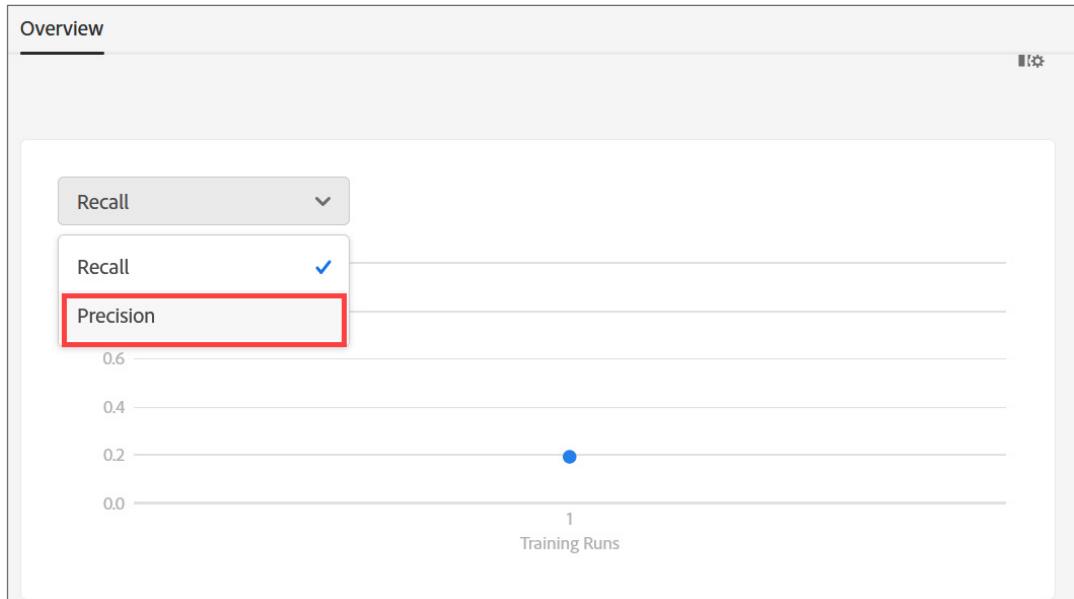
You should now see the **Overview** section at the top of the page, as shown, with two additional columns, **RECALL** and **PRECISION** beside the **STATUS** column.



The **RECALL** column displays the proportion of correct positive predictions (true positives) over the total number of cases that are actually positive (true positives + false negatives).

The **PRECISION** column displays the proportion of correct positive predictions (true positives) over all cases that are predicted as positive (true positives + false positives).

- To change the display of the metric in the graph, on the **Overview** tab, click the **Recall** drop-down arrow and select the **Precision** metric, as shown. Notice the change in the graph.



Training a model requires more than one run. All the training runs are visible on the **Models** page. You can compare run results and decide which one is the most successful run.

### Task 3: Score the machine learning model

In this task, you will score the model to generate product recommendations insights for customers.

- Ensure you are in the **Models** > Your recommendations page. For example, **Models** > **std121\_Recommendations**.
- Click **Training run 1**. The **Training run 1** page opens.
- At the upper-right corner of the page, click **Score**. The **Run Scoring** page opens.
- Click the **AEP Demo - Recommendations Input** option to select the input dataset. The **AEP Demo - Recommendations Input** pane is displayed on the right.
- Click **Next**. The **Run Scoring** page updates to display the **Scoring results** section.
- To select a dataset into which Experience Platform can render results, click the **AEP Demo - Recommendations Output** option. The **AEP Demo - Recommendations Output** pane is displayed on the right.
- Click **Next**. The **Run scoring** page updates to display the **Configuration** section. You can modify configuration parameters here. However, for this exercise, do not make any changes.

8. Click **Finish**. The **Training run 1** page opens. The page displays the **Scoring run 1** details. The status changes from **Pending** to **Complete** after a few minutes.

| SCORING RUN NAME | CREATED             | STATUS     |
|------------------|---------------------|------------|
| Scoring run 1    | 10/07/2022, 7:14 PM | ● Complete |

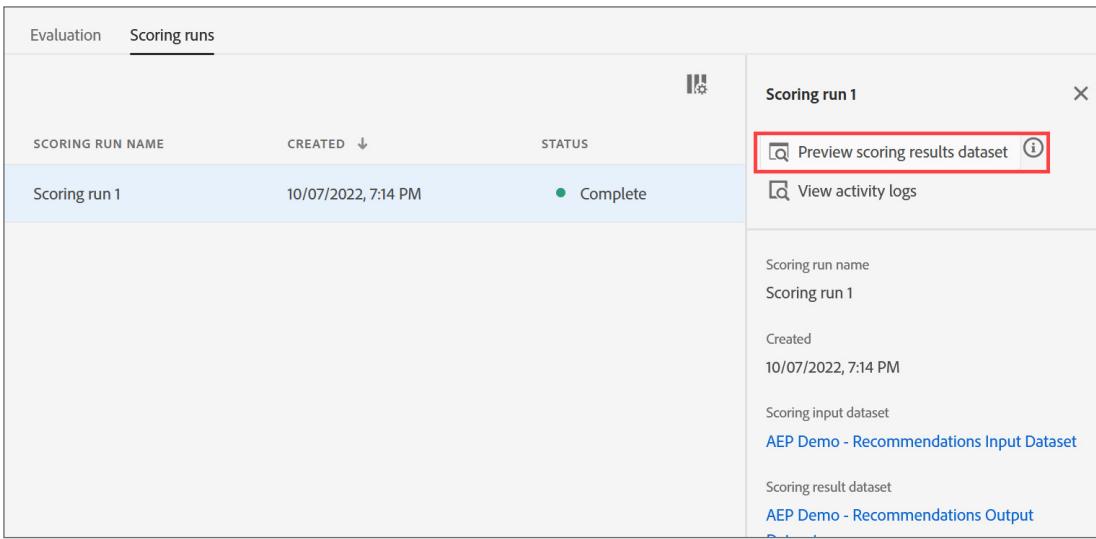
9. Click **Scoring run 1**. The **Scoring run 1** pane is displayed on the right.
10. In the **Scoring result dataset** section, click the link **AEP Demo - Recommendations Output**, as shown. The **AEP Demo - Recommendations Output** page opens.

| SCORING RUN NAME | CREATED             | STATUS     |
|------------------|---------------------|------------|
| Scoring run 1    | 10/07/2022, 7:14 PM | ● Complete |

Scoring run name  
Scoring run 1  
Created  
10/07/2022, 7:14 PM  
Scoring input dataset  
[AEP Demo - Recommendations Input Dataset](#)  
Scoring result dataset  
[AEP Demo - Recommendations Output Dataset](#)

11. Click **Preview dataset** at the upper-right corner. The preview dialog box for the dataset opens.
12. Observe the preview data and click **Close** to return to the **AEP Demo - Recommendations Output** page.
13. Click the browser back button to return to the **Training run 1** page.
14. Click **Scoring run 1**. The **Scoring run 1** pane opens on the right with the scoring run details.

15. Click **Preview scoring results dataset**, as shown. The **AEP Demo - Recommendations Output** dialog box opens with the scoring run details.



16. Observe the scoring results dataset and click **Close** to close the **AEP Demo - Recommendations Output** dialog box.

## Appendix

# Intelligent Services – Customer AI

### Introduction

Intelligent Services empower marketing analysts and practitioners to leverage the power of artificial intelligence and machine learning in customer experience use cases. This enables marketing analysts to set up predictions specific to a company's needs using business-level configurations without the need for data science expertise. Additionally, marketing practitioners can activate predictions in Adobe Experience Cloud, Adobe Experience Platform, and third-party applications.

Intelligent Services, built on Adobe Experience Platform, gives marketers responsible for customer experience access to AI-as-a-service, making it easy for anyone to predict customer behavior, measure the impact of a campaign, or ensure better return on every investment.

The following are some of the benefits of Intelligent Services:

- **Cost efficient speed to market:** Adobe has AI/ML expertise combined with deep CXM experience, and Intelligent Services have been designed specifically for marketers to get AI/ML off the ground immediately for common marketing use cases without the need to hire scarce and expensive data scientists, accelerating speed to market and actionable insights.
- **Higher accuracy in predictive insights:** Developed with the highest level of accuracy resultant through the adoption of advanced algorithmic AI/ML models, the ability to run AI/ML on a common dataset with the depth and granularity to enable slicing and dicing of predictions, and intelligence to provide the what and why behind predictive customer insights.
- **Fast time to insights and ROI:** Empower marketers to get started with AI/ML right away and start improving marketing ROI with new self-service workflows, custom configurations, and flexible options to operationalize predictive insights.

## Objectives

After completing this module, you will be able to:

- View schema and dataset configuration used for Intelligent Services
- Set up a new Customer AI instance
- Visualize propensity score and create segments

## Exercise 1: View Schema and Dataset Configuration

This exercise includes the following tasks:

1. View the schema used for Intelligent Services
2. View the dataset used for Intelligent Services

### Task 1: View the schema used for Intelligent Services

1. In the Experience Platform, ensure you are using the sandbox suggested by your instructor.
2. On the left navigation pane, under **DATA MANAGEMENT**, click **Schemas**.
3. Ensure you are on the **Browse** tab and type **AEP Demo - Intelligent Services - Customer Experience Event Schema**. The **AEP Demo - Intelligent Services - Customer Experience Event schema** is displayed.
4. Click the schema link. The **AEP Demo - Intelligent Services - Customer Experience Event Schema** page opens.
5. In the **Composition** pane on the left, observe the following:
  - a. The class of the schema is **XDM ExperienceEvent**. For the AEP Demo - Intelligent Services - Customer Experience Event Schema, XDM ExperienceEvent class is used by Intelligent Services.
  - b. Under **Field groups**, notice the **End User ID Detail** group. The End User ID Details field group is required for the main dataset that will store the data used by Intelligent Services.
  - c. Under **Field groups**, notice the **Consumer Experience Event** field group. The Consumer Experience Event field group is required for the main dataset that will store the data used by Intelligent Services.
  - d. Under **Identities**, observe the two Identities—the Primary identity has the path in the schema: `endUserIDs._experience.emailid.id` and the secondary identity has the path in the schema: `endUserIDs._experience.mcid.id`.
6. In the **Composition** pane, click the **End User ID Details** field group and notice the fields highlighted under **Structure**.
7. In the **Composition** pane, click the **Primary: endUserIDs.\_experience.emailid.id** identity first and then the **endUserIDs.\_experience.mcid.id** identity. Under **Structure**, notice that the field for both identities belong to the **End User ID Details** field group.

8. At the top of the **Composition** pane, under the **Schema** heading click on the name of the schema.
9. In the **Schema properties** pane on the right, note that **Profile** is toggled on for this schema.

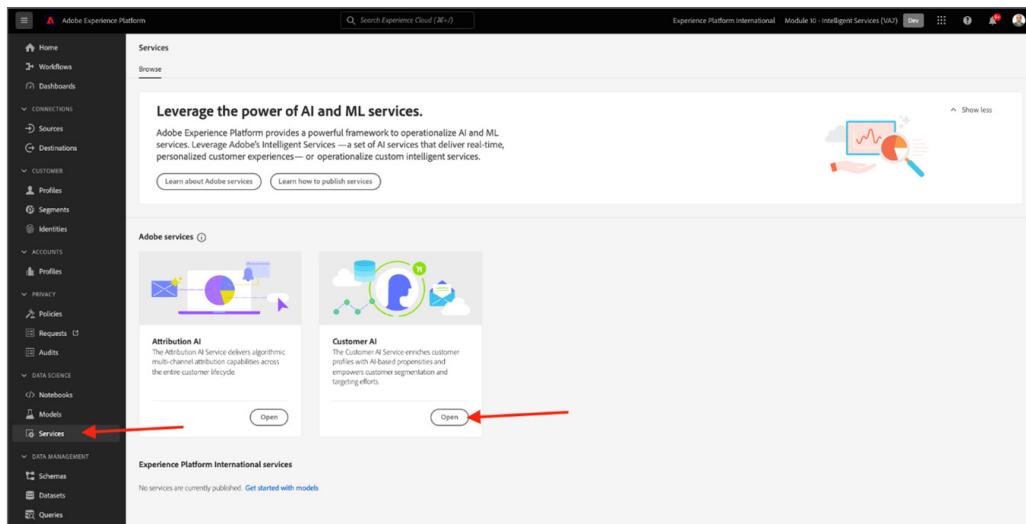
## Task 2: View Dataset needed for Intelligent Services

1. On the left navigation pane in Adobe Experience Platform, under **DATA MANAGEMENT**, click **Datasets**.
2. Ensure you are on the **Browse** tab, and type **AEP Demo - Intelligent Services - Customer Experience Event Dataset** in the search bar. The dataset appears.
3. Click the dataset link. The **AEP Demo - Intelligent Services - Customer Experience Event Dataset** page opens. Observe that data has already been ingested into the dataset. This is the data that will be used to generate predictive scores for individual profiles.
4. Click **Preview dataset** in the upper-right corner of the page. The dataset preview opens.
5. In the dialog that opens, explore the sample data that is displayed. Scroll through the data and click on the schema name on the left to display different fields and values in the dataset.
6. Click **Close** to close the preview dialog. The **AEP Demo - Intelligent Services - Customer Experience Event Dataset** page opens.
7. In the right pane, notice that **Profile** is toggled on.

## Exercise 2: Set up a New Customer AI Instance

Customer AI works by analyzing existing Consumer Experience Event data to predict churn or conversion propensity scores. Creating a new Customer AI instance enables marketeers to define goals and measures.

1. In Adobe Experience Platform, under **DATA SCIENCE** on the left menu, click **Services**. The **Services** page opens and displays all the available services.
2. On the **Customer AI** card, click **Open**, as shown. The **Customer AI** page opens.



3. Click **Create instance**. The **Create** page opens in the **Set up** tab.
4. Under **Basic information** section, update the following details:
  - a. Name: <student#\_date\_Product Purchase Propensity>. For example, student01\_113022\_Product Purchase Propensity
  - b. Description: Predict the likelihood for customers to purchase a product
  - c. Propensity type: Conversion
5. On the upper-right corner of the page, click **Save and Continue**.
6. In the **Select data** tab of the **Create** page, select the **AEP Demo - Intelligent Services - Customer Experience Event Dataset** checkbox and click **Add**. Observe the sample data displayed at the bottom of the page.

7. Click **Save and continue**. The **Define goal** tab is displayed.
8. Under the **Define your goal** section, click the **Select a dataset and field** drop-down menu and click **AEP Demo - Intelligent Services - Customer Experience Event Dataset**.
9. Select **commerce.purchases.value** as the target variable.
10. Click **Save and continue**.
11. Under the **Schedule** section, retain **Weekly** as the **Scoring frequency** and set the **Time** as close as possible to your current time.
12. Ensure that **Enable scores for Profile** is toggled on.
13. Click **Finish**. The **Create instance** dialog opens.
14. Click **Ok**. After you configure the instance, you can see it in the Customer AI instance list. You can view a summary of the setup and execution details by clicking on a blank area of the Customer AI instance row (not the blue link for the instance). The summary panel will also display error details in case errors have been found.

---

**Note:** You can modify any definition or attribute as long as your Customer AI instance's status is either **Awaiting training** or **Error**.

---

## Exercise 3: Visualize propensity score and create segments

Once your Customer AI instance completes a model run, it will enable you to visualize the propensity score that is evaluated to predict a customer performing a purchase in the next 30 days. The Customer AI dashboard shows the summary about score, distribution of population and the influential factors for the model to evaluate.

---

**Note:** Only a Customer AI instance with a status of Success will allow you to preview the service's insights.

---

This exercise includes the following tasks:

1. Predict propensity
2. Segment customers
3. View individual customer's score

### Task 1: Predict propensity

1. Ensure the **LAST RUN STATUS** indicates **Success** and click the instance you created in the previous exercise. The instance opens.
2. On the **Latest scores** tab, notice the scoring summary and distribution of scores.
3. Scroll down, if necessary, to see the groupings of low, medium, and high propensity.
4. Observe the count of profiles in each grouping.
5. Hover the cursor on the graphs for the **Top influential factors** to view the further breakdown of data distribution.

### Task 2: Segment customers

1. On the **Low Propensity** card, click **Create segment**. The segment builder page opens.
2. On the right pane, in the Name box, type a name. For example, <student#\_date\_Customer AI medium Propensity.
3. Click **Save** to save the segment definition. You can now use this segment for targeting using for instance Real-time CDP, Journey Orchestration and Adobe Target.

### Task 3: View individual customer's score

As the Customer AI propensity score becomes part of the Real-time Customer Profile, you can view individual customer's score.

1. In Adobe Experience Platform, under **CUSTOMER**, click **Profiles**.
2. Ensure you are on the **Browse** tab.
3. In the **Identity namespace** box, type and select **Email**.
4. In the **Identity value** box type an email ID. For example, type **smountain3hy@icq.com**. This email address is available in the ingested JSON file.
5. Click **View**. The profile is displayed.
6. Click the profile ID. The profile page opens.
7. On the **Attributes** tab, scroll down and check the propensity score calculated by the Customer AI model.

## Additional Resources

---

[Intelligent Services Overview](#)