

# DEP API Lab

## LAB Overview

Defining schemas and understanding how data is described by those schemas is one of the first things a customer will do on Adobe Experience Platform. This is the first step to ingesting data to the platform to gain marketing insights.

This lab will introduce you to the Adobe Experience Platform schema design user experience. You will learn how to create your own schemas, as well as browse existing components.

## Learning Objectives

**What should you walk away with after taking this Lab?**

- Create various AEP/XDM components using API Calls. Examples are:
  1. Class (Standard/Custom)
  2. Field Group (Standard/Custom)
  3. Data Type (Standard/Custom)
  4. Schema
  5. Descriptors
    - a. Identities (Primary/Secondary)
    - b. Relationship
    - c. Reference
- Adding some Schema configurations through Schema UI interface.
- Browse AEP/XDM Components through AEP Calls
- Validating the created Schemas through UI

## Pre Requisites

**Learners should have gone through the following concepts/courses/modules**

- Basics of XDM and Schema Composition
- Profile concepts and the important XDM components for Profile
- SID Methodology – Should have applied the SID methodology to the Customer Source Data Model and should have the expected XDM Data Model.
- Should be aware of the Source to XDM Mapping Document. This will be the source of all the Schemas created in this LAB.

## Lab Resources

Following artifacts should be accessible to the learners:

- Source to XDM Mapping document - [SID API Lab - Source to XDM Schema Mapping.xlsx](#)
- Postman collection to refer API calls – <Add link here >

## Lab Guide

- **Step 1- Create Custom Data Type**
  - Data Type Name - Profile Identities
    - Data types are very important XDM constructs as they allow us to define a complex structure once and use it multiple times.
  - API Call

POST https://platform.adobe.io/data/foundation/schemaregistry/tenant/datatypes

Params Authorization Headers (14) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "title": "Profile Identities",
3   "meta:extensible": true,
4   "description": "Profile Identities.",
5   "definitions": {
6     "profile-identities": {
7       "properties": {
8         "customerID": {
9           "title": "Customer ID",
10          "type": "string",
11          "description": "Customer Identifier."
12        },
13        "personalEmail": {
14          "title": "Personal Email",
15          "type": "string",
16          "description": "Personal Email Id."
17        }
18      }
19    }
20  },
21  "allOf": [
22    {
23      "$ref": "#/definitions/profile-identities"
24    }
25  ]
26 }
27 }
```

• Response

Body Cookies Headers (16) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "$id": "https://ns.adobe.com/dxp/datatypes/9faaf535c6b4b2f73f68960afebfdea8ce24dd8c6b74b7aa",
3   "meta:altId": " dxp.datatypes.9faaf535c6b4b2f73f68960afebfdea8ce24dd8c6b74b7aa",
4   "meta:resourceType": "datatypes",
5   "version": "1.0",
6   "title": "Profile Identities",
7   "type": "object",
8   "description": "Profile Identities.",
9   "definitions": {
10     "profile-identities": {
11       "properties": {
12         "customerID": {
13           "title": "Customer ID",
14           "type": "string",
15           "description": "Customer Identifier.",
16           "meta:xmType": "string"
17         },
18         "personalEmail": {
19           "title": "Personal Email",
20           "type": "string",
21           "description": "Personal Email Id.",
22           "meta:xmType": "string"
23         }
24       }
25     }
26   },
27   "allOf": [
```

• Sample Payload

## Data Type - Profile Identities

```
curl 'https://platform-int.adobe.io/data/foundation/schemaregistry/tenant/datatypes' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "title": "Profile Identities",
  "meta:extensible": true,
  "description": "Profile Identities.",
  "definitions": {
    "profile-identities": {
      "properties": {
        "customerID": {
          "title": "Customer ID",
          "type": "string",
          "description": "Customer Identifier."
        },
        "personalEmail": {
          "title": "Personal Email",
          "type": "string",
          "description": "Personal Email Id."
        }
      }
    }
  },
  "allof": [
    {
      "$ref": "#/definitions/profile-identities"
    }
  ]
}
```

- **Step 2- Create Custom field group**

- Field Group Name - Profile Identities [EE].
  - This field group will be used across all the experience event based schemas which will need to add this field group. This field group is used to mark the person identities (primary/secondary) in the schema. It gives the marketers a single place to find all the relevant identities defined for a Union View of all the schemas created from the experience event class.
- This field group needs to extend the Experience-Event class.
- API Call

The screenshot shows a REST client interface with a POST request to `https://platform.adobe.io/data/foundation/schemaregistry/tenant/mixins`. The request body is a JSON object defining a custom field group named "Profile Identities [EE]". The JSON includes a schema reference, title, type, meta:extensible, meta:abstract, meta:tags, meta:intendedToExtend, description, definitions, and an allof array.

```
1 {
2   "$schema": "http://json-schema.org/draft-06/schema#",
3   "title": "Profile Identities [EE]",
4   "type": "object",
5   "meta:extensible": true,
6   "meta:abstract": true,
7   "meta:tags": {
8     "industry": ["all"]
9   },
10  "meta:intendedToExtend": ["https://ns.adobe.com/xdm/context/experienceevent"],
11  "description": "Profile Identities for experience events.",
12  "definitions": {
13    "profile-identities": {
14      "properties": {
15        "dxp:profileIdentities": {
16          "title": "Profile identities",
17          "$ref": "${COMPONENTS_DATA_TYPES_IDENTITIES_ID}",
18          "description": "Profile Identities."
19        }
20      }
21    }
22  },
23  "allof": [
24    {
25      "$ref": "#/definitions/profile-identities"
26    }
27  ]
28 }
```

- Response

```

1  {
2    "$id": "https://ns.adobe.com/dxp/mixins/2338ee6d793604ce87084f6db776b680b49ee2a2424c546d",
3    "meta:altId": "_dxp.mixins.2338ee6d793604ce87084f6db776b680b49ee2a2424c546d",
4    "meta:resourceType": "mixins",
5    "version": "1.0",
6    "title": "Profile Identities [EE]",
7    "type": "object",
8    "description": "Profile Identities for expereince events.",
9    "definitions": {
10     "profile-identities": {
11       "properties": {
12         "dxp:profileIdentities": {
13           "title": "Profile identities",
14           "$ref": "https://ns.adobe.com/dxp/datatypes/9faaf535c6b4b2f73f68960afebfdea8ce24dd8c6b74b7aa",
15           "description": "Profile Identities.",
16           "type": "object",
17           "meta:xdmType": "object"
18         }
19       }
20     },
21     "allOf": [
22       {
23         "$ref": "#/definitions/profile-identities",
24         "type": "object",
25         "meta:xdmType": "object"
26       }
27     ]
28   }
29 }

```

- Sample Payload

Custom Field Group - Profile Identities

```

curl 'https://platform-int.adobe.io/data/foundation/schemaregistry/tenant/fieldgroups' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "$schema": "http://json-schema.org/draft-06/schema#",
  "type": "object",
  "meta:extensible": true,
  "meta:abstract": true,
  "meta:tags": {
    "industry": ["all"]
  },
  "meta:intendedToExtend": ["https://ns.adobe.com/xdm/context/experienceevent"],
  "description": "Profile Identities for expereince events.",
  "definitions": {
    "profile-identities": {
      "properties": {
        "dxp:profileIdentities": {
          "title": "Profile identities",
          "$ref": "<Please Get the Schema ID for the Profile Identities Data type>",
          "description": "Profile Identities."
        }
      }
    }
  },
  "allOf": [
    {
      "$ref": "<Include the local definition here>"
    }
  ],
  "meta:status": "stable"
}'

```

- Step 3- Create Schema for orders

- Schema Name - Orders

- The Orders schema is created using the Experience event class and the corresponding field groups. This Order Schema will be used to map all the order related payloads from the Customer source systems. Also note that this schema contains the superset of all the properties of orders from all the possible order sources (batch/streaming). The individual Source datasets will be mapped to the subsets of this schema as required.

- The schema needs to have:
  - Experience Event Class
  - Order Details field group
  - Profile Identities Field Group
- API Call

```
POST https://platform.adobe.io/data/foundation/schemaregistry/tenant/schemas

Params Authorization Headers (14) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 {
2   ... "allOf": [
3     ... {
4       ... "$ref": "https://ns.adobe.com/xdm/context/experienceevent"
5     },
6     ... {
7       ... "$ref": "https://ns.adobe.com/xdm/context/experienceevent-order-details"
8     },
9     ... {
10      ... "$ref": "https://ns.adobe.com/dxp/mixins/b923d7af029efecd7f6f578478caec461eb9923cf6171f19"
11    },
12  ],
13  "title": "Orders",
14  "description": "Orders"
15 }
16
```

- Response

```
Body Cookies Headers (16) Test Results
Pretty Raw Preview Visualize JSON

1 {
2   "$id": "https://ns.adobe.com/dxp/schemas/110c651e4430d68b9a101000378e3d9b0488bb50cbdebba",
3   "meta:allId": ".dxp.schemas.110c651e4430d68b9a101000378e3d9b0488bb50cbdebba",
4   "meta:resourceType": "schemas",
5   "version": "1.0",
6   "title": "Orders",
7   "type": "object",
8   "description": "Orders",
9   "allOf": [
10    {
11      "$ref": "https://ns.adobe.com/xdm/context/experienceevent",
12      "type": "object",
13      "meta:xdmType": "object"
14    },
15    {
16      "$ref": "https://ns.adobe.com/xdm/context/experienceevent-order-details",
17      "type": "object",
18      "meta:xdmType": "object"
19    },
20    {
21      "$ref": "https://ns.adobe.com/dxp/mixins/b923d7af029efecd7f6f578478caec461eb9923cf6171f19",
22      "type": "object",
23      "meta:xdmType": "object"
24    }
25  ],
26   "refs": [
27     "https://ns.adobe.com/dxp/mixins/b923d7af029efecd7f6f578478caec461eb9923cf6171f19".
28   ]
29 }
```

- Sample Payload

## Orders Schema

```
curl 'https://platform-int.adobe.io/data/foundation/schemaregistry/tenant/schemas' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "allof": [
    {
      "$ref": "https://ns.adobe.com/xdm/context/experienceevent"
    },
    {
      "$ref": "https://ns.adobe.com/xdm/context/experienceevent-order-details"
    },
    {
      "$ref": <Please get the schema ID for the third component here (Data type, Field Group
, Class ?) >
    }
  ],
  "title": "Orders",
  "description": "Orders"
}'
```

- **Step 4- Create Primary Identity for Orders Schema**

- Source Property - "personalEmail"

- The Primary identity will be used as the primary person identifier in the order event record.

- **API Call**

The screenshot shows an API client interface with a POST request to `https://platform.adobe.io/data/foundation/schemaregistry/tenant/descriptors`. The request body is a JSON object defining a primary identity for the 'personalEmail' property.

```
1 {
2   "@type": "xdm:descriptorIdentity",
3   "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas/6fb36d0cb830cbdf6a69a89aec5ab7c1705173bafc10656f",
4   "xdm:sourceVersion": 1,
5   "xdm:sourceProperty": "/_dxp/profileIdentities/personalEmail",
6   "xdm:namespace": "Email",
7   "xdm:property": "xdm:code",
8   "xdm:isPrimary": true
9 }
```

- **Response**

Body Cookies Headers (15) Test Results

Pretty Raw Preview Visualize JSON

```

1  {
2    "@id": "9dbb0002fe6fb959b499a820d35f68ba000921776e340c3a",
3    "@type": "xdm:descriptorIdentity",
4    "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas/6fb36d0cb830cbdf6a69a89aec5ab7c1705173bafc10656f",
5    "xdm:sourceVersion": 1,
6    "xdm:sourceProperty": "/_dxp/profileIdentities/personalEmail",
7    "imsOrg": "37E0399C61687C4E0A495E06@AdobeOrg",
8    "version": "1",
9    "xdm:namespace": "Email",
10   "xdm:property": "xdm:code",
11   "xdm:isPrimary": true,
12   "meta:containerId": "2a1c4985-170c-4bd6-9c49-85170c4bd64d",
13   "meta:sandboxId": "2a1c4985-170c-4bd6-9c49-85170c4bd64d",
14   "meta:sandboxType": "development"
15 }

```

- 
- Sample Payload

**Orders - Primary Identity**

```

curl 'https://platform-dev.adobe.io/data/foundation/schemaregistry/tenant/descriptors' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "@type": "xdm:descriptorIdentity",
  "xdm:sourceSchema": <Please get the required schema ID here>,
  "xdm:sourceVersion": 1,
  "xdm:sourceProperty": "/_dxp/profileIdentities/personalEmail",
  "xdm:namespace": "Email",
  "xdm:property": "xdm:code",
  "xdm:isPrimary": true
}'

```

- **Step 5- Create Secondary Identity for Orders Schema**
  - Source Property – “customerID”
    - The secondary identities are used to stitch profile fragments together across the data sets.
  - API Call

POST ▼ https://platform.adobe.io/data/foundation/schemaregistry/tenant/descriptors

Params Authorization Headers (14) **Body** ● Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▼

```

1 {
2   "@type": "xdm:descriptorIdentity",
3   "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas/6fb36d0cb830cbdf6a69a89aec5ab7c1705173bafc10656f",
4   "xdm:sourceVersion": 1,
5   "xdm:sourceProperty": "/_dxp/profileIdentities/customerID",
6   "xdm:namespace": "CustomerID",
7   "xdm:property": "xdm:code",
8   "xdm:isPrimary": false
9 }
10 |

```

## • Response

Body Cookies Headers (15) Test Results

Pretty Raw Preview Visualize **JSON** ▼

```

1 {
2   "@id": "8b689108a5126b93cf0422c4225b1cce09454f638f4750ea",
3   "@type": "xdm:descriptorIdentity",
4   "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas/2055da8b29eca1774fc78cd04f905c41345e2b435440a7c1",
5   "xdm:sourceVersion": 1,
6   "xdm:sourceProperty": "/_dxp/profileIdentities/customerID",
7   "imsOrg": "37E0399C61687C4E0A495E06@AdobeOrg",
8   "version": "1",
9   "xdm:namespace": "CustomerID",
10  "xdm:property": "xdm:code",
11  "xdm:isPrimary": false,
12  "meta:containerId": "656e7272-a148-4504-ae72-72a148350405",
13  "meta:sandboxId": "656e7272-a148-4504-ae72-72a148350405",
14  "meta:sandboxType": "development"
15 }

```

## • Sample Payload

### Orders - Secondary Identity

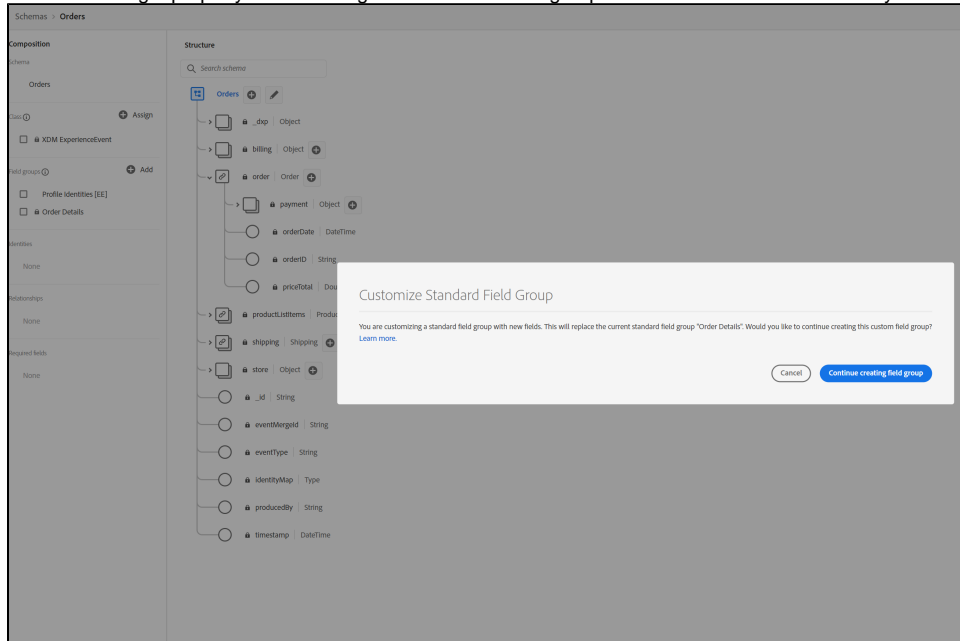
```

curl 'https://platform-dev.adobe.io/data/foundation/schemaregistry/tenant/descriptors' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "@type": "xdm:descriptorIdentity",
  "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas
/6fb36d0cb830cbdf6a69a89aec5ab7c1705173bafc10656f",
  "xdm:sourceVersion": 1,
  "xdm:sourceProperty": <Get the right property path for the source property here>,
  "xdm:namespace": "CustomerID",
  "xdm:property": "xdm:code",
  "xdm:isPrimary": <Should this be true or false ?>
}'

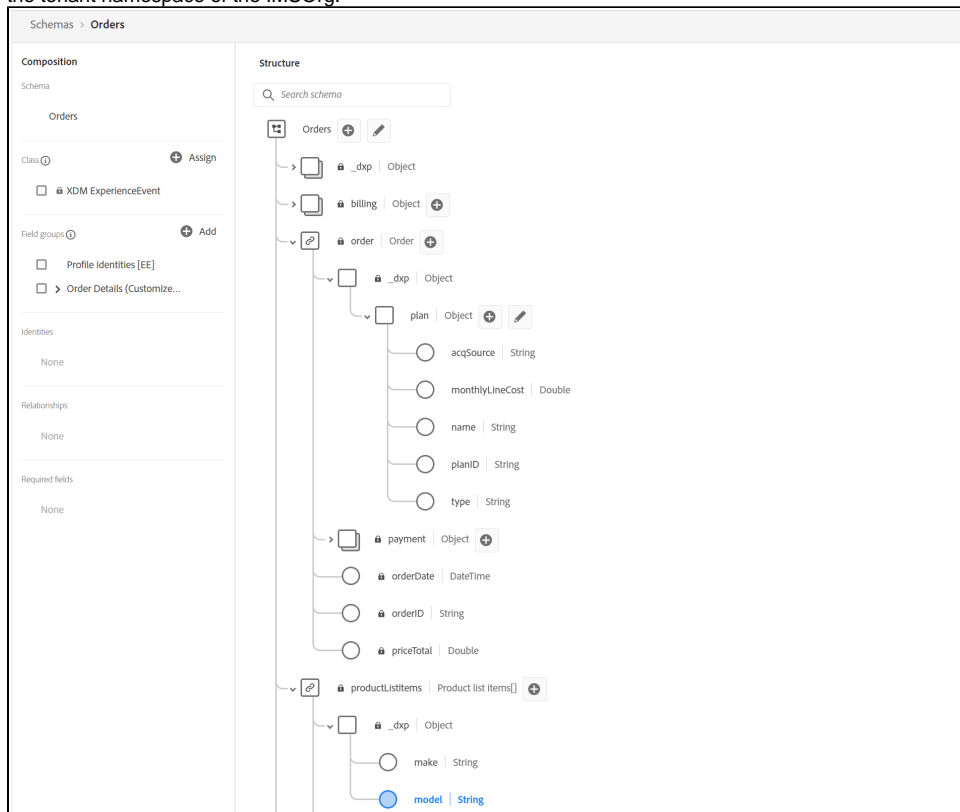
```



- **Go to the UI Schema browser**
  - Add Custom Properties
    - Adding a property to an existing standard XDM filed group creates an extended version in your sandbox.



- Please note below the added custom properties. Please note that each added property adds a top level nested object "\_dxp" which is the tenant namespace of the IMSOrg.



- **Step 6- Create Custom STORE Class**
  - Class Name - Store
    - The Store class is required to be created custom because there is no out of the box standard XDM Store Class.
    - The store class needs to be of the record behavior.
  - API Call

POST https://platform.adobe.io/data/foundation/schemaregistry/tenant/classes

Params Authorization Headers (14) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "title": "Store",
3   "type": "object",
4   "meta:extends": [
5     "https://ns.adobe.com/xdm/data/record"
6   ],
7   "description": "Store.",
8   "definitions": {
9     "store-class": {
10      "properties": {
11        "dxp:storeId": {
12          "title": "Store ID",
13          "type": "string",
14          "description": "Unique Identifier for the Store."
15        },
16        "dxp:storeName": {
17          "title": "Store Name",
18          "type": "string",
19          "description": "Store name."
20        }
21      }
22    }
23  },
24  "allOf": [
25    {
26      "$ref": "https://ns.adobe.com/xdm/data/record"
27    }
28  ]
29 }
```

## Response

Body Cookies Headers (16) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "$id": "https://ns.adobe.com/dxp/classes/e103d25ccfb8e6ebb0bf8e9ad7f55e506bc5c2a892d782ff",
3   "meta:altId": "_dxp.classes.e103d25ccfb8e6ebb0bf8e9ad7f55e506bc5c2a892d782ff",
4   "meta:resourceType": "classes",
5   "version": "1.0",
6   "title": "Store",
7   "type": "object",
8   "description": "Store.",
9   "definitions": {
10     "store-class": {
11       "properties": {
12         "dxp:storeId": {
13           "title": "Store ID",
14           "type": "string",
15           "description": "Unique Identifier for the Store.",
16           "meta:xdmType": "string"
17         },
18         "dxp:storeName": {
19           "title": "Store Name",
20           "type": "string",
21           "description": "Store name.",
22           "meta:xdmType": "string"
23         }
24       }
25     }
26   },
27   "allOf": [
28     {
29       "$ref": "https://ns.adobe.com/xdm/data/record"
30     }
31   ]
32 }
```

## Sample Payload

### Custom Store Class

```
curl 'https://platform-int.adobe.io/data/foundation/schemaregistry/tenant/classes' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "title": "Store",
  "type": "object",
  "meta:extends": [
    "https://ns.adobe.com/xdm/data/record"
  ],
  "description": "Store.",
  "definitions": {
    "store-class": {
      "properties": {
        "dxp:storeID": {
          "title": "Store ID",
          "type": "string",
          "description": "Unique Identifier for the Store."
        },
        "dxp:storeName": {
          "title": "Store Name",
          "type": "string",
          "description": "Store name."
        }
      }
    }
  },
  "allOf": [
    {
      "$ref": "https://ns.adobe.com/xdm/data/record"
    },
    {
      "$ref": "#/definitions/store-class"
    }
  ]
}
```

- **Step 7- Create Custom Store Details field group**

- Field Group Name - Store Details.
  - Store details field group is required to have additional store attributes. These were not added to the Store Class to keep it thin and extendable by other store related field groups (if required) in future.
- This field group needs to extend the "Store" class.
- API Call

```
POST https://platform.adobe.io/data/foundation/schemaregistry/tenant/mixins

Params Authorization Headers (14) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 {
2   "title": "Store Details",
3   "type": "object",
4   "meta:intendedToExtend": [
5     "https://ns.adobe.com/dxp/classes/e103d25ccfb8e6ebb0bf8e9ad7f55e506bc5c2a892d782ff"
6   ],
7   "description": "Store Details.",
8   "definitions": {
9     "store-details": {
10      "properties": {
11        "dxp:storeInagurationDate": {
12          "title": "Store Inaguration Date",
13          "type": "string",
14          "format": "date-time",
15          "description": "Store inaguration date(example - 2018-11-13T20:20:39+00:00).",
16        },
17        "dxp:storeOpenTime": {
18          "title": "Store Open Time",
19          "type": "string",
20          "format": "time",
21          "description": "Store Open Time(example - 08:00:00+00:00).",
22        },
23        "dxp:storeCloseTime": {
24          "title": "Store Close Time",
25          "type": "string",
26          "format": "time",
27          "description": "Store close time.(example - 20:00:00+00:00).",
28        }
29      }
30    }
31  }
```

• Response

```
Body Cookies Headers (16) Test Results
Pretty Raw Preview Visualize JSON

1 {
2   "$id": "https://ns.adobe.com/dxp/mixins/40b61d79b232a2f3136bd0b5e1768cb9a66b2bce44646368",
3   "meta:altId": "_dxp.mixins.40b61d79b232a2f3136bd0b5e1768cb9a66b2bce44646368",
4   "meta:resourceType": "mixins",
5   "version": "1.0",
6   "title": "Store Details",
7   "type": "object",
8   "description": "Store Details.",
9   "definitions": {
10     "store-details": {
11       "properties": {
12         "dxp:storeInagurationDate": {
13           "title": "Store Inaguration Date",
14           "type": "string",
15           "format": "date-time",
16           "description": "Store inaguration date(example - 2018-11-13T20:20:39+00:00).",
17           "meta:xdmType": "date-time"
18         },
19         "dxp:storeOpenTime": {
20           "title": "Store Open Time",
21           "type": "string",
22           "format": "time",
23           "description": "Store Open Time(example - 08:00:00+00:00).",
24           "meta:xdmType": "string"
25         },
26         "dxp:storeCloseTime": {
27           "title": "Store Close Time",
28         }
29       }
30     }
31  }
```

• Sample Payload

### Custom Store Details Field Group

```
curl 'https://platform-int.adobe.io/data/foundation/schemaregistry/tenant/mixins' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "title": "Store Details",
  "type": "object",
  "meta:intendedToExtend": [
    <Get the schema ID for the Store class>
  ],
  "description": "Store Details.",
  "definitions": {
    "store-details": {
      "properties": {
        "dxp:storeInagurationDate": {
          "title": "Store Inaguration Date",
          "type": "string",
          "format": "date-time",
          "description": "Store inaguration date(example - 2018-11-13T20:20:39+00:00).",
        },
        "dxp:storeOpenTime": {
          "title": "Store Open Time",
          "type": "string",
          "format": "time",
          "description": "Store Open Time(example - 08:00:00+00:00).",
        },
        "dxp:storeCloseTime": {
          "title": "Store Close Time",
          "type": "string",
          "format": "time",
          "description": "Store close time.(example - 20:00:00+00:00).",
        },
        "dxp:address": {
          "title": "Store Address",
          "$ref": "https://ns.adobe.com/xdm/common/address",
          "description": "Store address."
        }
      }
    }
  },
  "allof": [
    {
      "$ref": "#/definitions/store-details"
    }
  ]
}
```

- **Step 8- Create Schema for Store**

- Schema Name - Store
  - Store Schema will serve as a Look up entity for the Orders Schema. In order to filter on certain Store attributes while qualifying segments, the Orders schema will have to join to this store schema to access store related attributes.
  - The Orders schema will have to establish a relationship with the Store schema to join to it. That is done through the relationship descriptors (Shown in further steps).
- The schema needs to have:
  - "Store" Class
  - "Store Details" field group
- API Call

POST

https://platform.adobe.io/data/foundation/schemaregistry/tenant/schemas

Params

Authorization

Headers (14)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

1

{

2

"type": "object",

3

"title": "Store Schema",

4

"description": "Store Schema.",

5

"allOf": [

6

{

7

"\$ref": "https://ns.adobe.com/dxp/classes/e103d25ccfb8e6ebb0bf8e9ad7f55e506bc5c2a892d782ff"

8

},

9

{

10

"\$ref": "https://ns.adobe.com/dxp/mixins/40b61d79b232a2f3136bd0b5e1768cb9a66b2bce44646368"

11

}

12

]

13

}

14

- 
- Response

Body

Cookies

Headers (16)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

"\$id": "https://ns.adobe.com/dxp/schemas/1ad53f4aca94fddb86b3a1ef070d0ff2e3007eb55807410e",

"meta:altId": "\_dxp.schemas.1ad53f4aca94fddb86b3a1ef070d0ff2e3007eb55807410e",

"meta:resourceType": "schemas",

"version": "1.0",

"title": "Store Schema",

"type": "object",

"description": "Store Schema.",

"allOf": [

{

"\$ref": "https://ns.adobe.com/dxp/classes/e103d25ccfb8e6ebb0bf8e9ad7f55e506bc5c2a892d782ff",

"type": "object",

"meta:xdmType": "object"

},

{

"\$ref": "https://ns.adobe.com/dxp/mixins/40b61d79b232a2f3136bd0b5e1768cb9a66b2bce44646368",

"type": "object",

"meta:xdmType": "object"

}

],

"refs": [

"https://ns.adobe.com/dxp/mixins/40b61d79b232a2f3136bd0b5e1768cb9a66b2bce44646368",

"https://ns.adobe.com/dxp/classes/e103d25ccfb8e6ebb0bf8e9ad7f55e506bc5c2a892d782ff"

],

"imsOrg": "37E0399C61687C4E0A495E06@AdobeOrg",

"meta:extractable": false,

"meta:abstract": false.

- 
- Sample Payload

## Store Schema

```
curl 'https://platform-int.adobe.io/data/foundation/schemaregistry/tenant/schemas' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "type": "object",
  "title": "Store Schema",
  "description": "Store Schema.",
  "allof": [
    {
      "$ref": "<Get the Schema ID for the Store Class>"
    },
    {
      "$ref": "<Get the Schema ID for the Store Details Field Group>"
    }
  ]
}
```

### • Step 9- Create Store Namespace

- A namespace is required to be created as it provides some context to a Schema identity. Each Identity in AEP needs to be associated to a certain namespace.
- API Call

POST <https://platform.adobe.io/data/core/idnamespace/identities>

Params Authorization Headers (14) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "name": "Store",
3   "code": "Store",
4   "description": "storeID",
5   "idType": "NON_PEOPLE"
6 }
```

### • Response

Body Cookies Headers (15) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "updateTime": 1658251742884,
3   "code": "Store",
4   "status": "ACTIVE",
5   "description": "storeID",
6   "id": 12272336,
7   "createTime": 1658251742884,
8   "idType": "NON_PEOPLE",
9   "namespaceType": "Custom",
10  "name": "Store",
11  "custom": true
12 }
```

### • Sample Payload

## Store Namespace

```
curl 'https://platform-dev.adobe.io/data/core/idnamespace/identities' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "name": "Store",
  "code": "Store",
  "description": "storeID",
  "idType": "NON_PEOPLE"
}'
```

- **Step 10- Create Primary Identity for Store Schema**

- Source Property - "storeID"

- Primary identity for Store is required as it is used as the storage key Store entity in the Profile database.
    - When a relationship is defined from an event schema (Orders) to a lookup schema (Store), it assumes that the entity column referred in the lookup table is defined as a Primary Identity.

- API Call

POST <https://platform.adobe.io/data/foundation/schemaregistry/tenant/descriptors>

Params Authorization Headers (14) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "@type": "xdm:descriptorIdentity",
3   "xdm:sourceSchema": "https://ns.adobe.com/dxp/classes/c5171a6313de98c92e152a692925a3039b18ad3ddce6f5e3",
4   "xdm:sourceVersion": 1,
5   "xdm:sourceProperty": "/_dxp/storeID",
6   "xdm:namespace": "storeID",
7   "xdm:property": "xdm:code",
8   "xdm:isPrimary": true
9 }
10
```

- Response

Body Cookies Headers (13) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "type": "http://ns.adobe.com/aep/errors/XDM-1810-400",
3   "title": "Descriptor validation error",
4   "status": 400,
5   "report": {
6     "registryRequestId": "da91c1b9-9029-4093-bf2a-a74077255d30",
7     "timestamp": "07-19-2022 05:32:08",
8     "detailed-message": "The namespace ID storeID is not valid.",
9     "sub-errors": []
10  },
11   "detail": "The namespace ID storeID is not valid."
12 }
```

- Sample Payload



### Store - Primary Identity

```
curl 'https://platform-dev.adobe.io/data/foundation/schemaregistry/tenant/descriptors' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "@type": "xdm:descriptorIdentity",
  "xdm:sourceSchema": "https://ns.adobe.com/dxp/classes
/c5171a6313de98c92e152a692925a3039b18ad3ddce6f5e3",
  "xdm:sourceVersion": 1,
  "xdm:sourceProperty": "/_dxp/storeID",
  "xdm:namespace": "storeID",
  "xdm:property": "xdm:code",
  "xdm:isPrimary": true
}
```

- **Step 11- Create Relationship Identity Descriptors for Order Schema**

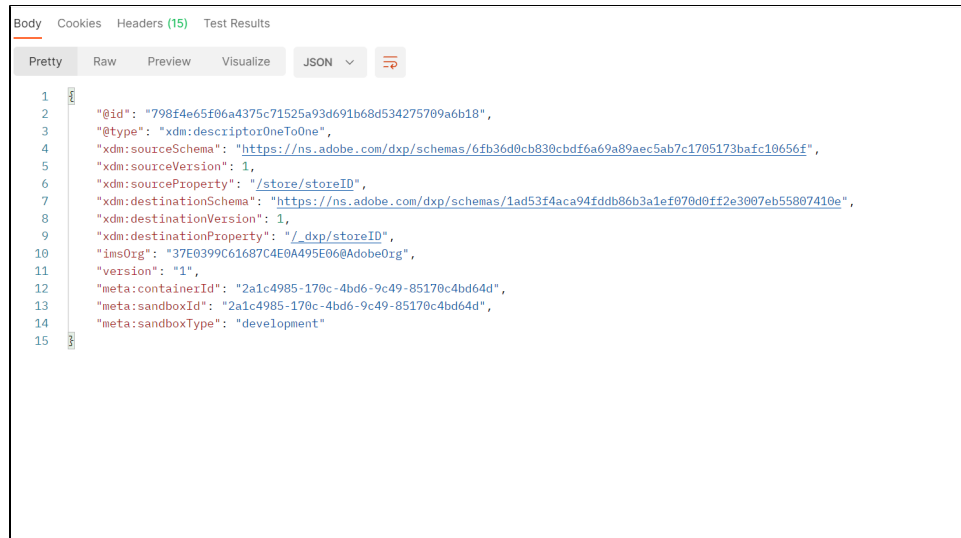
- Relationship descriptors provides a mechanism in AEP for two schemas to talk to each other. Relationships are defined from either the Profile schemas or the Event schemas. An event schema needs to have explicit relationships defined to all its associated lookup schemas.
- Create Descriptors for relations
  - **Step 11a - "Orders" : "Store"**
    - Orders to Store is a M:1 (Many to 1) relationship which implies that a store could have multiple orders placed but an order record would only belong to a single store.

- **API Call**

The screenshot shows an API client interface with a POST request to `https://platform.adobe.io/data/foundation/schemaregistry/tenant/descriptors`. The request body is a JSON object defining a relationship descriptor between 'Orders' and 'Store' schemas.

```
1 {
2   "@type": "xdm:descriptorOneToOne",
3   "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas/6fb36d0cb830cbdf6a69a89aec5ab7c1705173bafc10656f",
4   "xdm:sourceVersion": 1,
5   "xdm:sourceProperty": "/store/storeID",
6   "xdm:destinationSchema": "https://ns.adobe.com/dxp/schemas/1ad53f4aca94fddb86b3a1ef070d0ff2e3007eb55807410e",
7   "xdm:destinationVersion": 1,
8   "xdm:destinationProperty": "/_dxp/storeID"
9 }
10
11 |
```

- **Response**



- Sample Payload

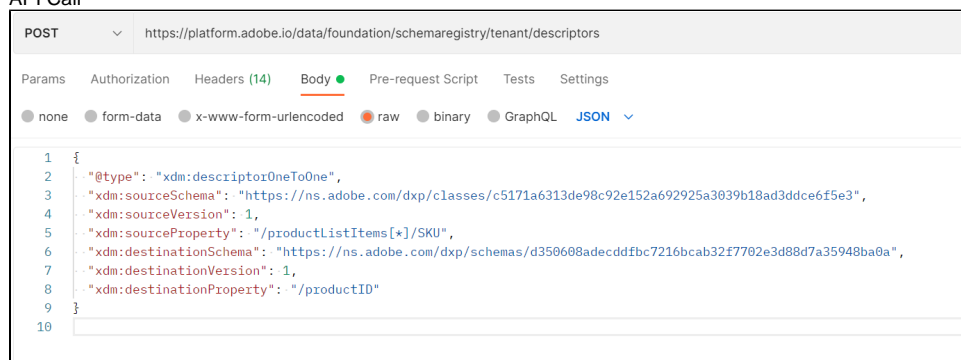
### Orders=>Store - Relationship Descriptor

```

curl 'https://platform-dev.adobe.io/data/foundation/schemaregistry/tenant/descriptors' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "@type": "xdm:descriptorOneToOne",
  "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas
/6fb36d0cb830cbdf6a69a89aec5ab7c1705173bafc10656f",
  "xdm:sourceVersion": 1,
  "xdm:sourceProperty": "/store/storeID",
  "xdm:destinationSchema": "https://ns.adobe.com/dxp/schemas
/1ad53f4aca94fddb86b3a1ef070d0ff2e3007eb55807410e",
  "xdm:destinationVersion": 1,
  "xdm:destinationProperty": "/_dxp/storeID"
}'

```

- **Step 11b - "Orders" : "Product"**
  - Orders to Product is a M:1 (Many to 1) relationship which implies that a Product could be part of multiple orders placed but an order record would only have a single product.
  - One thing to note here is that this relationship is actually M:N as per the source system but since the "Order-Product" bridge table is de-normalized onto the Orders table as an array of "Product List Items", this is now a M:1 relationship with multiple Products stored as array elements on a single order record.
- API Call



- Response

```
Body Cookies Headers (13) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "type": "http://ns.adobe.com/xdp/errors/XDM-1015-408",
3   "title": "Descriptor validation error",
4   "status": 408,
5   "report": {
6     "registryRequestId": "ac98be75-260f-4195-a6e9-a2af41ce49d",
7     "timestamp": "07-19-2022 05:34:17",
8     "detailed-message": "The path [/productListItems[*]/_id] defined in xdm:sourceProperty could not be found in the source schema https://ns.adobe.com/dxp/classes/c517a6313de9b02e152e492925a3079b18a3ddedf5e3",
9     "sub-errors": []
10  },
11   "detail": "The path [/productListItems[*]/_id] defined in xdm:sourceProperty could not be found in the source schema https://ns.adobe.com/dxp/classes/c517a6313de9b02e152e492925a3079b18a3ddedf5e3."
12 }
```

- 
- Sample Payload

### Orders=>Product - Relationship Descriptor

```
curl 'https://platform-dev.adobe.io/data/foundation/schemaregistry/tenant/descriptors' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "@type": "xdm:descriptorOneToOne",
  "xdm:sourceSchema": "https://ns.adobe.com/ddgxdmstg/schemas
/db398e8046ab0f76c381cc18938def19e24c39e72c3c7e4b",
  "xdm:sourceVersion": 1,
  "xdm:sourceProperty": "/productListItems[*]/_id",
  "xdm:destinationSchema": "<Product Schema ID>",
  "xdm:destinationVersion": 1,
  "xdm:destinationProperty": "/productID"
}'
```

- Step 11c - "Orders" : "Plan"
  - Orders to Plan is a M:1 (Many to 1) relationship which implies that a Plan could be part of multiple orders placed but an order record would only be associated to a single plan.
- API Call

```
POST https://platform.adobe.io/data/foundation/schemaregistry/tenant/descriptors
Params Authorization Headers (14) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
1 {
2   "@type": "xdm:descriptorOneToOne",
3   "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas/110c651e4430d68b9a101000378e3d9b0488bb50cbdebba",
4   "xdm:sourceVersion": 1,
5   "xdm:sourceProperty": "/order/_dxp/plan/planID",
6   "xdm:destinationSchema": "https://ns.adobe.com/dxp/schemas/17a76a0ad31add7bbf7c6afc651c75b01add14d0cdd332e8",
7   "xdm:destinationVersion": 1,
8   "xdm:destinationProperty": "/planID"
9 }
```

- 
- Response

```
Body Cookies Headers (15) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "id": "ee89c31ae5d848c44a21769486ed5da9ccb9ed842efc37ca",
3   "@type": "xdm:descriptorOneToOne",
4   "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas/110c651e4430d68b9a101000378e3d9b0488bb50cbdebba",
5   "xdm:sourceVersion": 1,
6   "xdm:sourceProperty": "/order/_dxp/plan/planID",
7   "xdm:destinationSchema": "https://ns.adobe.com/dxp/schemas/17a76a0ad31add7bbf7c6afc651c75b01add14d0cdd332e8",
8   "xdm:destinationVersion": 1,
9   "xdm:destinationProperty": "/planID",
10  "imsOrg": "37E0399C61687C4E0A495E06@AdobeOrg",
11  "version": "1",
12  "meta:containerId": "656e7272-a148-4504-ae72-72a148350405",
13  "meta:sandboxId": "656e7272-a148-4504-ae72-72a148350405",
14  "meta:sandboxType": "development"
15 }
```

- 
- Sample Payload

## Orders=>Plan- Relationship Descriptor

```
curl 'https://platform-dev.adobe.io/data/foundation/schemaregistry/tenant/descriptors' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "@type": "xdm:descriptorOneToOne",
  "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas
/110c651e4430d68b9a101000378e3d9b0488bb50cbdebba",
  "xdm:sourceVersion": 1,
  "xdm:sourceProperty": "/order/_dxp/plan/planID",
  "xdm:destinationSchema": "https://ns.adobe.com/dxp/schemas
/17a76a0ad31add7bbf7c6afc651c75b01add14d0cdd332e8",
  "xdm:destinationVersion": 1,
  "xdm:destinationProperty": "/planID"
}'
```

### • Step 12 - Create Reference Identity Descriptors for Order Schema

- Reference Identity descriptors are required to be created on the reference keys on the Schema. Since Order schema needs to define relationships to all its look up schemas, we would need to define a reference identity each for all the look up reference keys (Product, Store, Plan) in the order schema.
- Please note that this step is not required when you create schemas through the UI as these reference identities are created automatically on the backend once you define a relationship.

#### • Step 12a - Store Reference Descriptor

##### • API Call

POST <https://platform.adobe.io/data/foundation/schemaregistry/tenant/descriptors>

Params Authorization Headers (14) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "@type": "xdm:descriptorReferenceIdentity",
3   "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas/6fb36d0cb830cbdf6a69a89aec5ab7c1705173bafc10656f",
4   "xdm:sourceVersion": 1,
5   "xdm:sourceProperty": "/store/storeID",
6   "xdm:identityNamespace": "Store"
7 }
8
```

##### • Response

Body Cookies Headers (15) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "@id": "2008b2dc4671f65322624d649fa3168d2e88d254d62da8ed",
3   "@type": "xdm:descriptorReferenceIdentity",
4   "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas/6fb36d0cb830cbdf6a69a89aec5ab7c1705173bafc10656f",
5   "xdm:sourceVersion": 1,
6   "xdm:sourceProperty": "/store/storeID",
7   "imsOrg": "37E0399C61687C4E0A495E06@AdobeOrg",
8   "version": "1",
9   "xdm:identityNamespace": "Store",
10  "meta:containerId": "2a1c4985-170c-4bd6-9c49-85170c4bd64d",
11  "meta:sandboxId": "2a1c4985-170c-4bd6-9c49-85170c4bd64d",
12  "meta:sandboxType": "development"
13 }
```

##### • Sample Payload

## Orders - Reference Identity Descriptor for Store

```
curl 'https://platform-dev.adobe.io/data/foundation/schemaregistry/tenant/descriptors' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "@type": "xdm:descriptorReferenceIdentity",
  "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas
/6fb36d0cb830cbdf6a69a89aec5ab7c1705173bafc10656f",
  "xdm:sourceVersion": 1,
  "xdm:sourceProperty": "/store/storeID",
  "xdm:identityNamespace": "Store"
}'
```

- **Step 12b - Product Reference Descriptor**
- **API Call**

POST <https://platform.adobe.io/data/foundation/schemaregistry/tenant/descriptors>

Params Authorization Headers (14) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded **raw** binary GraphQL JSON

```
1 {
2   "@type": "xdm:descriptorReferenceIdentity",
3   "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas/2055da8b29eca1774fc78cd04f905c41345e2b435440a7c1",
4   "xdm:sourceVersion": 1,
5   "xdm:sourceProperty": "/productListItems[*]/SKU",
6   "xdm:identityNamespace": "productID"
7 }
```

- **Response**

Body Cookies Headers (15) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "@id": "d9c7fe61919660f7587b22f18e0eae01178025f0013e21b0",
3   "@type": "xdm:descriptorReferenceIdentity",
4   "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas/2055da8b29eca1774fc78cd04f905c41345e2b435440a7c1",
5   "xdm:sourceVersion": 1,
6   "xdm:sourceProperty": "/productListItems[*]/SKU",
7   "imsOrg": "37E0399C61687C4E0A495E06@AdobeOrg",
8   "version": "1",
9   "xdm:identityNamespace": "productID",
10  "meta:containerId": "656e7272-a148-4504-ae72-72a148350405",
11  "meta:sandboxId": "656e7272-a148-4504-ae72-72a148350405",
12  "meta:sandboxType": "development"
13 }
```

- **Sample Payload**

## Orders - Reference Identity Descriptor for Product

```
curl 'https://platform-dev.adobe.io/data/foundation/schemaregistry/tenant/descriptors' \
-H 'Authorization: Bearer <TOKEN>' \
-H 'x-api-key: acp_ui_platform' \
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \
-H 'x-sandbox-name: prod' \
-H 'Content-Type: application/json' \
-X POST \
--data-raw '{
  "@type": "xdm:descriptorReferenceIdentity",
  "xdm:sourceSchema": "https://ns.adobe.com/ddgxdmstg/schemas
/db398e8046ab0f76c381cc18938def19e24c39e72c3c7e4b",
  "xdm:sourceVersion": 1,
  "xdm:sourceProperty": "/productListItems[*]/_id",
  "xdm:identityNamespace": ""
}'
```

- **Step 12c - Plan Reference Descriptor**
- **API Call**

POST https://platform.adobe.io/data/foundation/schemaregistry/tenant/descriptors

Params Authorization Headers (14) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "@type": "xdm:descriptorReferenceIdentity",
3   "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas/110c651e4430d68b9a101000378e3d9b0488bb50cbdebba",
4   "xdm:sourceVersion": 1,
5   "xdm:sourceProperty": "/order/_dxp/plan/planID",
6   "xdm:identityNamespace": "planID"
7 }
```

- **Response**

Body Cookies Headers (15) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "@id": "62228f92a3dab0c651e4430d68b9a101000378e3d9b0488bb50cbdebba",
3   "@type": "xdm:descriptorReferenceIdentity",
4   "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas/110c651e4430d68b9a101000378e3d9b0488bb50cbdebba",
5   "xdm:sourceVersion": 1,
6   "xdm:sourceProperty": "/order/_dxp/plan/planID",
7   "imsOrg": "37E0399C61687C4E0A495E06@AdobeOrg",
8   "version": "1",
9   "xdm:identityNamespace": "planID",
10  "meta:containerId": "656e7272-a148-4504-ae72-72a148350405",
11  "meta:sandboxId": "656e7272-a148-4504-ae72-72a148350405",
12  "meta:sandboxType": "development"
13 }
```

- **Sample Payload**

## Orders - Reference Identity Descriptor for Plan

```
curl 'https://platform-dev.adobe.io/data/foundation/schemaregistry/tenant/descriptors' \  
-H 'Authorization: Bearer <TOKEN>' \  
-H 'x-api-key: acp_ui_platform' \  
-H 'x-gw-ims-org-id: 37E0399C61687C4E0A495E06@AdobeOrg' \  
-H 'x-sandbox-name: prod' \  
-H 'Content-Type: application/json' \  
-X POST \  
--data-raw ' {  
  "@type": "xdm:descriptorReferenceIdentity",  
  "xdm:sourceSchema": "https://ns.adobe.com/dxp/schemas/  
/110c651e4430d68b9a101000378e3d9b0488bb50cbdebba",  
  "xdm:sourceVersion": 1,  
  "xdm:sourceProperty": "/order/_dxp/plan/planID",  
  "xdm:identityNamespace": "planID"  
}  
'
```