Deliver offers using the Batch Decisioning API

With the Batch Decisioning API, you can populate a dataset with the best offers for all profiles in an Adobe Experience Platform segment for a decision scope. For example, an organization may want to run batch decisioning so they can send offers to a message delivery vendor. Those offers are then used as content that is sent out for batch message delivery to the same segment of users.

To do this, the organization would:

- Run the Batch Decisioning API, which contains two requests:
    1. A **Batch GET request** to get batch workload status.
    2. A **Batch POST request** to start a workload to batch process offer selections.
- Export the dataset to the message delivery vendor API.

Getting started {#getting-started}

Before using this API, make sure you complete the following pre-requisite steps.

Prepare the decision

Follow the steps below to prepare your decisions:

- In order to export a decision result, create a dataset using the "ODE Decision Events" schema.
- Create a Platform segment which should be evaluated and then updated. Refer to the segmentation documentation to learn more about how to update segment membership evaluation.
- Create a decision (which has a decision scope that consists of a Decision ID and a Placement ID) in Adobe Journey Optimizer. Refer to the section on defining decision scopes of the guide on creating decisions to learn more.

API requirements

All Batch Decisioning requests require the following additional headers:

- `Content-Type`: `application/json`
- `x-request-id`: A unique string that identifies the request.
- `x-sandbox-name`: The sandbox name.
- `x-sandbox-id`: The sandbox ID.

Retrieve information on a batch decision {#retrieve-information-on-a-batch-decision}

To retrieve information on a specific decision, make a GET request to the `/workloads/decisions` endpoint while providing the corresponding workload ID value for your decision.

API format

```
GET  {ENDPOINT_PATH}/{CONTAINER_ID}/workloads/decisions/{WORKLOAD_ID}
```

Parameter	Description	Example
{ENDPOINT_PATH}	The endpoint path for repository APIs.	https://platform.adobe.io/data/core/ode
{CONTAINER_ID}	The container where the decisions are located.	e0bd8463-0913-4ca1-bd84-6309134ca1f6
{WORKLOAD_ID}	The UUID generated by Offer Decisioning that identifies a single workload.	47efef25-4bcf-404f-96e2-67c4f784a1f5

Request

```
curl -X GET 'https://platform.adobe.io/data/core/ode/0948b1c5-fff8-3b76-
ba17-909c6b93b5a2/workloads/decisions/f395ab1f-dfaf-48d4-84c9-
199ad6354591' \
-H 'x-request-id: 7832a42a-d4e5-413b-98e8-e49bef056436' \
-H 'Content-Type: application/json' \
-H 'x-api-key: {API_KEY}' \
-H 'x-gw-ims-org-id: {IMS_ORG}' \
-H 'x-sandbox-name: {SANDBOX_NAME}' \
-H'x-sandbox-id: {SANDBOX_ID}' \
-H 'Authorization: {ACCESS_TOKEN}'
```

Response

```
{
    "@id": "f395ab1f-dfaf-48d4-84c9-199ad6354591",
    "xdm:imsOrgId": "{IMS_ORG}",
    "xdm:containerId": "0948b1c5-fff8-3b76-ba17-909c6b93b5a2",
    "ode:createDate": 1648076994405,
    "ode:status": "COMPLETED"
}
```

Property	Description	Example
@id	The UUID generated by Offer Decisioning that identifies a single workload.	5d0ffb5e-dfc6-4280-99b6-0bf3131cb8b8
xdm:imsOrgId	The IMS Organization ID	9GTO98D5F@AdobeOrg

Property	Description	Example
`xdm:containerId`	The Container ID	`0948b1c5-fff8-3b76-ba17-909c6b93b5a2`
`ode:createDate`	The time when Decision Workload request was created.	`1648076994405`
`ode:status`	The status of the workload starts with "QUEUED" and changes to "PROCESSING", "INGESTING", "COMPLETED" or "ERROR".	`ode:status: "COMPLETED"`
`ode:statusDetail`	This shows more details such as sparkJobId and batchID if the status is "PROCESSING" or "INGESTING". It shows the error details if the status is "ERROR".	

Start a batch process {#start-a-batch-process}

To start a workload to batch process decisions, make a POST request to the `/workloads/decisions` endpoint.

API format

```
POST {ENDPOINT_PATH}/{CONTAINER_ID}/workloads/decisions
```

Parameter	Description	Example
`{ENDPOINT_PATH}`	The endpoint path for repository APIs.	`https://platform.adobe.io/data/core/ode`
`{CONTAINER_ID}`	The container where the decisions are located.	`e0bd8463-0913-4ca1-bd84-6309134ca1f6`

Request

```
curl -X POST 'https://platform.adobe.io/data/core/ode/0948b1c5-fff8-3b76-
ba17-909c6b93b5a2/workloads/decisions' \
-H 'x-request-id: f671a589-eb7b-432f-b6b9-23d5b796b4dc' \
-H 'Content-Type: application/json' \
-H 'x-api-key: {API_KEY}' \
-H 'x-gw-ims-org-id: {IMS_ORG}' \
-H 'x-sandbox-name: {SANDBOX_NAME}' \
-H 'x-sandbox-id: {SANDBOX_ID}' \
-H 'Authorization: {ACCESS_TOKEN}' \
-d '{
  "xdm:segmentIds": [
    "609028e4-e66c-4776-b0d9-c782887e2273"
```

```
    ],
    "xdm:dataSetId": "6196b4a1a63bd118dafe093c",
    "xdm:propositionRequests": [
          {
              "xdm:activityId": "xcore:offer-activity:1410cdcda196707b",
              "xdm:placementId": "xcore:offer-placement:1410c4117306488a",
              "xdm:itemCount": 1
          }
    ],
    "xdm:includeContent": false
  }'
```

Property	Description	Example
`xdm:segmentIds`	The unique identifier of the segment. It can only contain one value.	`609028e4-e66c-4776-b0d9-c782887e2273`
`xdm:dataSetId`	The output dataSet that decision events can be written into.	`6196b4a1a63bd118dafe093c`
`xdm:propositionRequests`	A wrapper that contains the placementId and activityId	
`xdm:activityId`	The unique identifier of the decision.	`xcore:offer-activity:1410cdcda196707b`
`xdm:placementId`	The unique placement identifier.	`xcore:offer-placement:1410c4117306488a`
`xdm:itemCount`	This is an optional field showing the number of items such as options requested for the decisioning scope. By default, the API returns one option per scope, but you can explicitly ask for more options by specifying this field. A minimum of 1 and a maximum of 30 options can be requested per scope.	`1`
`xdm:includeContent`	This is an optional field and is `false` by default. If `true`, the offer content is included in the decision events of dataset.	`false`

Refer to the Decision Management documentation for an overview of the main concepts.

Response

```
{
    "@id": "47efef25-4bcf-404f-96e2-67c4f784a1f5",
    "xdm:imsOrgId": "9GTO98D5F@AdobeOrg",
    "xdm:containerId": "0948b1c5-fff8-3b76-ba17-909c6b93b5a2",
    "ode:createDate": 1648078924834,
    "ode:status": "QUEUED"
}
```

Property	Description	Example
`@id`	The UUID generated by Offer Decisioning that identifies a single workload.	`5d0ffb5e-dfc6-4280-99b6-0bf3131cb8b8`
`xdm:imsOrgId`	The ID for your IMS Organization.	`9GTO98D5F@AdobeOrg`
`xdm:containerId`	Your container ID.	`0948b1c5-fff8-3b76-ba17-909c6b93b5a2`
`ode:createDate`	The time when the decision workload request was created.	`1648078924834`
`ode:status`	The status of the workload.	`ode:status: "QUEUED"`

Service levels {#service-levels}

The end-to-end time for every batch decision is the duration from the time the workload is created to the decision result available in output dataset. The segment size in the POST request payload is the main factor that affects the end-to-end batch decision time. The below are some observations for different segment sizes:

- segment size <= 10K: 6 minutes
- segment size <= 1M: 10 minutes
- segment size <= 15M: 75 minutes

Limitations {#limitations}

The following limitations exist with Batch Decisioning API:

- **Single batch job**: Currently, only a single batch job can be run per dataset at a time. Any other requests would respond with HTTP 429 (Too Many Requests).
- **Frequency capping**: A batch runs off of the profile snapshot that occurs once a day. The Batch Decisioning API caps the frequency and always loads profiles from the most recent snapshot.

Next steps {#next-steps}

By following this API guide, you have checked for the workload status and delivered offers using the [!DNL Batch Decisioning] API. For more information, see the overview on Decision Management.