Create an Omnichannel Experience

Regardless of the channel that you want to serve an experience on (mobile app, server-side integration for Web, etc.), the process for creating an Engine API experience is the same with the exception of the actions available for the experience.

Follow these steps to configure an Engine API experience.

1. Click **EXPERIENCES** in the top navigation bar, and then select **Web**.
2. Click the arrow to the right of **CREATE EXPERIENCE**, and then select **Omnichannel**.
3. Name the experience.
4. Configure the WHY settings.
    a. Click **WHY**.
    b. Select an experience type and then click **NEXT**.
    c. Select a goal metric and then click **NEXT**. Refer to Event/Metric Mapping in Target, Event, & Condition Mapping for a table of default metrics and their respective associated Engine API events.
    d. Configure as necessary the secondary metrics, and then click **NEXT**.
    e. Confirm that the WHY summary statement is accurate, and then click **SAVE**.
5. Configure the WHO settings. Refer to Target Type Mapping in Target, Event, & Condition Mapping for a table of target types and their respective associated Engine API events.
    a. Click **WHO** and then click **ADD TARGET**.
    b. Click a category on the Target Type panel, select a target, and then configure it as necessary. See WHO Targets in Omnichannel Experiences in this documentation for more information.
6. Configure the WHAT settings.
    a. Click **WHAT** and then click **ADD ACTION**.
    b. Click a category on the Action Type panel, select an action template, and then configure it as necessary. See Available Omnichannel Action Types for more information about the action templates typically available for Engine API clients. See Using Action Conditions in this documentation for guidance on configuring action conditions.
7. Configure the WHEN settings.
    a. Click **WHEN** and then click **START AND STOP TIME**.
    b. Configure the start and, optionally, stop date and time as well as dayparting and time zone settings as necessary, and then click **SAVE**. See Configure Experience Timing for more information.
8. Click **PREVIEW** to test the experience. See Previewing an Experience for more information.
9. Click **ACTIVATE** when you're ready to deploy the experience.

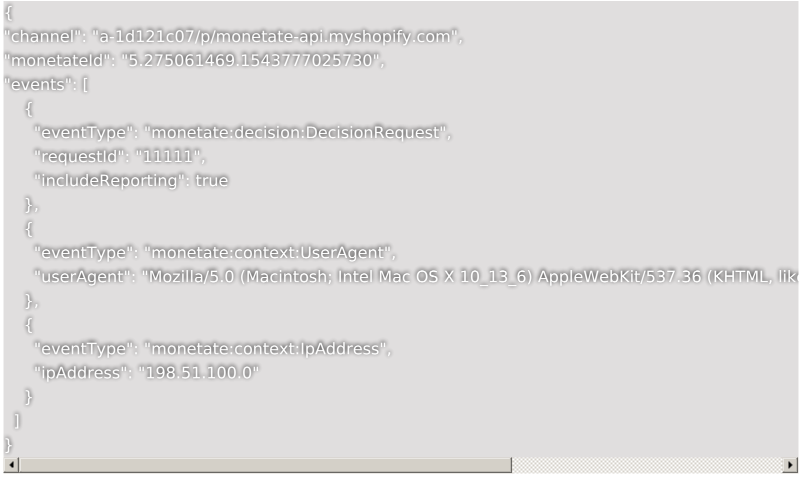
See Calling the Engine API for more information about Engine API request calls for Omnichannel experiences.

WHO Targets in Omnichannel Experiences

Engine API experiences support most WHO targets that you can use in an experience built for a JavaScript tag implementation, with the exception of a few that rely on data specific to Monetate tag–based integrations.

In Monetate tag implementations, many of the targets are evaluated based on information passed via the request header from the browser (information such as IP address, used to determine the location and related information, and user agent, from which device type, and operating system are derived).

In a server-side implementation, these are not available to Monetate and must be provided in the body of the Engine API request, as shown in this example.



```
{
"channel": "a-1d121c07/p/monetate-api.myshopify.com",
"monetateId": "5.275061469.1543777025730",
"events": [
    {
      "eventType": "monetate:decision:DecisionRequest",
      "requestId": "11111",
      "includeReporting": true
    },
    {
      "eventType": "monetate:context:UserAgent",
      "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, lik
    },
    {
      "eventType": "monetate:context:IpAddress",
      "ipAddress": "198.51.100.0"
    }
  ]
}
```

Using Action Conditions

Similar to JavaScript tag-based implementations, you can apply action conditions to the actions you configure in Engine API experiences.

For example, if you want an action to appear on a particular page type, click **ADD CONDITION**, expand the options of the **Page type** category, and then select **Page type =**.

Type the page type into the field to complete the action condition. The page type is defined by the `setPageType` data you pass to the platform.

Here is the request body with `pageType` specified in the `PageView` event:

```
{
"channel": "a-1d121c07/p/monetate-api.myshopify.com",
"monetateId": "5.275061469.1543777025730",
"events": [
    {
      "eventType": "monetate:decision:DecisionRequest",
      "requestId": "11111",
      "includeReporting": true
    },
    {
      "eventType": "monetate:context:PageView",
      "pageType": "home",
      "url": "test/foo"
    }
  ]
}
```

Refer to Action Condition Mapping in Target, Event, & Condition Mapping for a table of action condition options and their respective associated Engine API events.

Previewing an Experience

The experience preview has two functionalities:

- **Experience Preview**: This preview mimics experience evaluation in full so anyone who views it is evaluated as if they are a normal site visitor. To qualify for the experience, you must satisfy any targets and action conditions associated with the experience. The Engine API also decides what split you should see and then serves it.
- **Variant Preview**: This preview allows you to see the content returned by a specific variant. Anyone who views it sees the actions associated with the chosen variant when they satisfy the action conditions. Variant preview disregards any targets set in the experience.

The experience preview loads in a new browser window or tab. The URL is typically for the account's home page and includes a `preview` query parameter. The value for the parameter is an encoded preview token.

Here is an example of a preview URL:

```
http://www.monetate.com/?
preview=6.3.eJyrVkpMTs4vzSuJz0xRsjI1tjTTUUpOB7HNzczNjcyBPBAnrzQnR0cptaJAycrQ1NDAwtjc2NRQRym9CCRQCwDQbBOe.xKAg
```

In a JavaScript tag-based implementation, the Monetate tag recognizes the `preview` parameter, adds it to its own call to the Monetate platform, and then applies the relevant actions to preview.

However, for Engine API integrations you must provide the preview token to the Engine API call.

> The preview token is valid for 1 hour.

Here is a code sample for passing the `preview` parameter.

```
{
 "preview: "preview token",
 "events": [
   {
    "pageType": "home",
    "eventType": "monetate:context:PageView"
   }
 ]
}
```