Available Omnichannel Actions

Omnichannel action templates are used with the Engine API as well as for other usages. Contact your dedicated Services team members to request Omnichannel action templates be added to your account or to scope potential custom action template needs.

- All Omnichannel actions have the option for managed impressions. Action templates that support managed impressions or slot filtering are added individually.

---

**Omni Data Collect**

This action type is analogous to the Data Collect or Do Nothing actions used in a traditional Monetate tag-based integration.

<table>
<thead>
<tr>
<th>WHAT</th>
<th>ACTION TYPE</th>
<th>OTHER</th>
<th>OMNI DATA COLLECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Omni Data Collect</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Used for A/A tests and other data-collection scenarios where no visible action should be shown to the user. 

*This action contains no built-in targeting.*

**Optional Inputs**

Apply this action when the following conditions are met:

---

It returns no content and is intended for use in the following scenarios:
- A/A tests
- Data collections
- **Standard Test experiences** when you want a single control group, which you can accomplish by removing the native control, adding a subsequent split, and then assigning a Data Collect action with identical action conditions

**Omni JSON**

This action type is the standard Omni action type. The template has a single required JSON input into which you can add any properly formed JSON that's subsequently returned within the Engine API response.

This action can include optional inputs, such as an **Element Selector** or **Component ID** field or the **ADD CONDITION** selector for adding action conditions.

**Common uses of this action type include the following:**

- **As a feature flag**: The experiment variant could return the feature flag as **ON** or **TRUE**, and the control variant could return the feature flag as **OFF** or **FALSE**
- **As a component ID**: Each variant could have a separate component ID, and the visitor sees the assigned variant associated with the component ID
- **As a path to hosted content**: Each variant might have a separate content path, and the visitor sees the content from the path associated with the assigned variant
- **As a path to CMS content**: The client fetches the CMS content and displays it to the visitor

Another possibility for this action type is to use it with snippets of HTML.

**Omni HTML**
Instead of a required JSON input, this action type has a required HTML code editor that offers both a WYSIWYG view and a source code view.

**Omni HTML**

Returns HTML

*This action contains no built-in targeting.*

**Required Inputs**

**HTML String**

**Optional Inputs**

**Element Selector**

Selector for the element which will receive the html

**Apply this action when the following conditions are met:**

[ADD CONDITION]

This action template can include optional inputs, such as an **Element Selector** field or the **ADD CONDITION** selector for adding action conditions.

**Omni Full-Page Redirect**

This action type is similar to Full-Page Test experiences or the URL Redirect action in Monetate tag-based integrations. When a **DecisionRequest** event that doesn't include slot filtering is submitted, the customer might qualify for redirects actions, non-redirect actions, or some combination of the two.
If the visitor qualifies for a redirect action and then is assigned to the experiment group, only that action is returned in the response. If the customer qualifies for multiple redirect actions and is assigned to the experiment group for each of them, then the experience with the highest priority is returned. If the visitor doesn’t qualify for any redirect actions or is assigned to the control group for all redirect actions that they should qualify for, then all non-redirect actions are returned.

Single-request full-page redirects also work when the DecisionRequest event includes the managedImpressions=True attribute. If the customer qualifies for a redirect action and is assigned to the experiment group, then only that action is returned in the response. The action includes an impressionId token, but the response indicates that it is a control action with the flag isControl=true.

The impressionId token for a given redirect action must be reported back to Monetate regardless of whether the customer is in the experiment or control group. However, the visitor should be redirected to the URL included in the action if isControl=False.

**Omni Intelligent Recommendations**

This action type allows you to leverage Monetate’s Product Recommendations feature with the Engine API.
Configure the recommendation strategies, set the minimum, maximum, and pinned products, and then the Engine API responds with an array of product details.

The requesting application is responsible for creating the container or carousel and all desired styling for the product recommendations.

**Product Catalog Custom Fields in Recommendations Actions**

You can use custom fields (for example, star ratings) from a product catalog in an Omni Intelligent Recommendations.

Product IDs must match across the Monetate JavaScript API, the product catalog, and the request.

The requesting application is responsible for creating the container or carousel and all desired styling for the product recommendations.
Any custom field that you use in the action must also appear in the product catalog configured as part of the recommendation strategy that you selected for the action.

**Intelligent Recommendations Action Best Practices**

If your site uses a hybrid integration, keep in mind these best practices when configuring an Omni Intelligent Recommendations action:

- You only have to send product IDs if you base recommendations on a specific product (for example, a recommendation strategy configured with the Viewed and Also Viewed recommendation algorithm based on the last item viewed in any session or the first item viewed on the current page).
- **ProductDetailView**, **CartLines**, and **PurchaseLines** events are obtained from the Monetate tag. You're not required to send these.

If your site uses a pure Engine API implementation, keep in mind that you must send all **ProductDetailView**, **CartLines**, and **PurchaseLines** events.

When selecting recommendation strategies for this type of action, be aware of the following conditions related to the recommendation algorithm configured as part of the recommendation strategy:

- **Most Viewed (Product Detail Page), Top Selling by Purchase Count, and Top Selling by Gross Revenue**: You don't need to send any product-specific information in the requests.
- **Viewed and Also Viewed based on the first item viewed on the current page or on the last item viewed in any session**: Send the **ProductDetailView** event in the request.
- **Viewed and Also Viewed based on item(s) viewed on the current page**: Send all product IDs in the **ProductThumbnailView** event.

**Examples of Requests for Product Recommendation Strategies**

Include any relevant context.

This example is for recommendation strategies that use the Most Viewed (Product Detail Page), Top Selling by Purchase Count, or Top Selling by Gross Revenue recommendation algorithms.
This example is for recommendation strategies that use the Viewed and Also Viewed recommendation algorithm with recommendations based on the first item viewed on the current page or the last item viewed in any session.
This example is for recommendation strategies that use the Viewed and Also Viewed and the Viewed and Later Purchased recommendation algorithms based on the last item purchased in any session.

You must send the product ID of the last purchase.
This example is for recommendation strategies that use the Viewed and Also Viewed and the and Later Purchased recommendation algorithms based on the last item carted in any session.

You must send the product ID of the last item added to the cart.
Product Recommendations Reporting

When using the Engine API for recommendations, you must make subsequent requests when products are visually presented to the customer and when a customer clicks or taps any of the recommended products. This request is necessary to record the applicable item impressions and item clicks.

- Use `monetate:record:RecImpressions` whenever products are presented to the customer or whenever a customer swipes along a carousel to see additional recommendations.
- Use `monetate:record:RecClicks` whenever a customer clicks or taps a product recommendation so that Monetate can record the click.

See the Engine API documentation within the Monetate platform for code examples for using `monetate:record:RecImpressions` and `monetate:record:RecClicks`.

Complete the following steps to report product recommendations.

1. In the initial request, send the relevant product IDs depending on the recommendation strategy that you’re using.

2. Find the unique recToken in the initial response. The response contains a unique recToken for each product within the recommendation strategy for that experience, such as the one in this example:

   "recToken": "rt.1.WzL4LjI3NTQsiCiwNDkwNTMiLCAiLDQ5LDUzIlwgMTQ4ODYsIDAsIClyMDIxLTExLTA0VDExOjQ4OjyLjExNyM2NVoiXQ"

3. Send the recToken in a second request when the impression occurs, followed by a third request when the click occurs.

   - **Impressions**

     ```json
     {
     "eventType": "monetate:record:RecImpressions",
     "recImpressions": [
     "rt.1.WzL4LjI3NTQsiCiwNDkwNTMiLCAiLDQ5LDUzIlwgMTQ4ODYsIDAsIClyMDIxLTExLTA0VDExOjQ4OjyLjExNyM2NVoiXQ",
     "rt.1.yyy"
     ]
     }
     ```

   - **Clicks**

     ```json
     {
     "eventType": "monetate:record:RecClicks",
     "recClicks": [
     "rt.1.WzL4LjI3NTQsiCiwNDkwNTMiLCAiLDQ5LDUzIlwgMTQ4ODYsIDAsIClyMDIxLTExLTA0VDExOjQ4OjyLjExNyM2NVoiXQ",
     "rt.1.yyy"
     ]
     }
     ```