Recommendations AI (https://cloud.google.com/recommendations/)
Documentation (https://cloud.google.com/recommendations-ai/docs/) Guides

User events

Beta

This product is in a pre-release state and might change or have limited support. For more information, see the product launch stages (https://cloud.google.com/products/#product-launch-stages).

This page describes the user event object, including listing potential user event types, and provides sample data for all user event types.

For help with recording user events, see Recording user events (https://cloud.google.com/recommendations-ai/docs/record-events).

User event types

There are several user event types that you can record as users browse your retail site:

User event name	User action
add-to-cart (#add-to-cart)	Adds product to cart
add-to-list (#add-to-list)	Adds product to list
category-page-view (#category-page-view)	Views special pages, such as sale or promotion pages
checkout-start (#checkout-start)	Starts checkout process
detail-page-view (#detail-page-view)	Views product detail page
home-page-view (#home-page-view)	Views homepage
page-visit (#page-visit)	Views generic page not included in other page view events
purchase-complete (#purchase-complete)	Completes checkout
refund (#refund)	Returns or is refunded for a purchased product

User event name	User action
remove-from-cart (#remove-from-cart)	Removes product from cart
remove-from-list (#remove-from-list)	Removes product from list
search (#search)	Searches the catalog
shopping-cart-page-view (#shopping-cart-page-view)	Views shopping cart

For detailed information about the UserEvent object, see UserEvent (https://cloud.google.com/recommendations-ai/docs/reference/rest/v1beta1/projects.locations.catalogs.eventStores.userEvents) .

Event type priority

For the highest quality recommendations, we recommend that that you record user events for all event types. The following table describes the priority of the different user event types. You must log the highest priority user events to achieve quality data models.

Priority	User Events
Required for initial live experiment	`add-to-cart` (#add-to-cart) `checkout-start` (#checkout-start) `detail-page-view` (#detail-page-view) `home-page-view` (#home-page-view) `purchase-complete` (#purchase-complete)
Important for improving model quality over time	`category-page-view` (#category-page-view) `remove-from-cart` (#remove-from-cart) `search` (#search) `shopping-cart-page-view` (#shopping-cart-page-view)
Nice to have	`add-to-list` (#add-to-list)

`page-visit` (#page-visit)

`refund` (#refund)

`remove-from-list` (#remove-from-list)

Example data for all User Event Types

The following are examples of the data format for each event type supported by Recommendations AI.

For all examples, `user-id` is optional. The catalog item information fields (`currencyCode`, `originalPrice`, `displayPrice`, and `stockState`) are optional; if they are provided, they override the value in the Catalog. The `experimentIds` field is needed only if you are running an A/B experiment (https://cloud.google.com/recommendations-ai/docs/a-b-testing).

The `recommendation-token` field is optional; it is used to measure recommendation performance (https://cloud.google.com/recommendations-ai/docs/recommender). Learn more (https://cloud.google.com/recommendations-ai/docs/recommendation-tokens).

For more details about the User Event object, see User Events (https://cloud.google.com/recommendations-ai/docs/reference/rest/v1beta1/projects.locations.catalogs.eventStores.userEvents) .

Add to cart

The following shows an example of the `add-to-cart` user event format.

The `productDetails` section is required, and it must include the `quantity` field. If you include the pricing information, the `currencyCode` field is also required. The `cartId` field is optional.

JAVASCRIPT PIXEL TAG MANAGER

```
var user_event = {
  "eventType": "add-to-cart",
  "userInfo": {
    "visitorId": "visitor-id",
    "userId": "user-id"
  },
  "eventDetail": {
    "recommendationToken": "recommendation-token",
    "experimentIds": "experiment-group"
  },
  "productEventDetail": {
    "cartId" : "5678",
    "productDetails": [{
      "id": "123",
      "currencyCode": "USD",
      "originalPrice": 140.00,
      "displayPrice": 130.00,
      "quantity": "12"
    }]
  }
}
```

Add to list

The following example shows the `add-to-list` user event data format.

The `listId`, `productDetails`, and `quantity` fields are required.

JAVASCRIPT PIXEL   TAG MANAGER

```
var user_event = {
  "eventType": "add-to-list",
  "userInfo": {
    "visitorId": "visitor-id",
    "userId": "user-id"
  },
  "eventDetail": {
   "experimentIds": "experiment-group"
  },
  "productEventDetail": {
    "productDetails": [{
      "id": "123",
      "quantity":2
      "currencyCode": "USD",
      "originalPrice": 140.00,
      "displayPrice": 130.00,
    }],
    "listId" : "thanksgiving gift ideas"
   }
}
```

Category page view

The following example shows the `category-page-view` user event data format.

The `pageCategories` field is required. While there is usually only one category associated with a page, the `pageCategories` field also supports a category hierarchy, which you can provide as a list.

JAVASCRIPT PIXEL TAG MANAGER

```
 var user_event = {
    "eventType": "category-page-view",
    "userInfo": {
      "visitorId": "visitor-id",
      "userId": "user-id"
    },
    "eventDetail": {
    "experimentIds": "experiment-group"
    },
    "productEventDetail": {
      "pageCategories": [{
        "categories": ["Sales", "Black Friday 2018"]
      }]
    }
}
```

Checkout start

The following example shows the `checkout-start` user event data format.

You must include a `productDetail` object for each product being purchased.

The `cartId` field is optional. The `purchaseTransaction` object is optional; if you include it, you must include the `revenue` and `currencyCode` fields. The `quantity` field is optional.

JAVASCRIPT PIXEL TAG MANAGER

```
 var user_event = {
   "eventType": "checkout-start",
   "userInfo": {
     "visitorId": "visitor-id",
     "userId": "user-id"
   },
   "eventDetail": {
    "experimentIds": "experiment-group"
   },
   "productEventDetail": {
      "cartId" : "5678",
      "productDetails": [{
        "id": "123",
        "originalPrice": 140.00,
        "displayPrice": 130.00,
        "currencyCode": "USD",
        "quantity":2
      },
      {
        "id": "567",
        "originalPrice": 32.00,
        "displayPrice": 54.00,
        "currencyCode": "USD",
        "quantity":1
      }],
      "purchaseTransaction": {
        "id": "143",
        "revenue": 28.34,
        "taxes": {"state": 3.4, "local": .41},
        "costs": {"manufacturing": 45.5, "cost": 12.4},
        "currencyCode": "USD"
    }
  }
}
```

Detail page view

The following example shows the `detail-page-view` user event data format.

The `productDetails` field is required. In most cases, `productDetails` contains details for the associated product, unless a bundle of items are being sold together.

If the view was provided from a recommendation, including the recommendation token provides data for model evaluation. Learn more (https://cloud.google.com/recommendations-ai/docs/recommendation-tokens).

JAVASCRIPT PIXEL TAG MANAGER

```
var user_event = {
  "eventType": "detail-page-view",
  "userInfo": {
    "visitorId": "visitor-id",
    "userId": "user-id"
  },
  "eventDetail": {
    "recommendationToken": "recommendation-token",
    "experimentIds": "experiment-group"
  },
  "productEventDetail": {
    "productDetails": [{
      "id": "123",
    }]
  }
}
```

Home page view

The following example shows the `home-page-view` user event data format.

JAVASCRIPT PIXEL TAG MANAGER

```
var user_event = {
  "eventType": "home-page-view",
  "userInfo": {
    "visitorId": "visitor-id",
    "userId": "user-id"
  },
}
```

Page Visit

The following example shows the `page-visit` user event data format.

The `productDetails` and `eventDetails` objects are optional.

JAVASCRIPT PIXEL TAG MANAGER

```
 var user_event = {
  "eventType": "page-visit",
  "userInfo": {
    "visitorId": "visitor-id",
    "userId": "user-id"
  },
  "eventDetail": {
    "experimentIds": "experiment-group"
  },
  "productEventDetail": {
    "productDetails": [{
      "id": "123"
    }]
  }
}
```

Purchase complete

The following shows an example of the `purchase-complete` user event data format.

The `purchaseTransaction` object is required, and must contain the `revenue` and `currencyCode` fields. The `quantity` field is required. The `cartId` field is optional.

JAVASCRIPT PIXEL TAG MANAGER

```
 var user_event = {
  "eventType": "purchase-complete",
  "userInfo": {
    "visitorId": "visitor-id",
    "userId": "user-id"
  },
  "eventDetail": {
    "experimentIds": "experiment-group"
  },
  "productEventDetail": {
    "cartId" : "5678",
    "productDetails": [{
      "id": "123",
      "originalPrice": 140.00,
      "displayPrice": 130.00,
      "quantity":2
      }],
    "purchaseTransaction": {
       "id": "143",
        "revenue": 28.34,
        "taxes": {"state": 3.4, "local": .41},
        "costs": {"manufacturing": 45.5, "cost": 12.4},
        "currencyCode": "USD"
     }
  }
}
```

Refund

The following example shows the `refund` user event data format.

The `productDetails` section is required, and must include the `id` and `quantity` fields.

JAVASCRIPT PIXEL TAG MANAGER

```
 var user_event = {
  "eventType": "refund",
  "userInfo": {
    "visitorId": "visitor-id",
    "userId": "user-id"
  },
  "eventDetail": {
    "experimentIds": "experiment-group"
  },
  "productEventDetail": {
    "productDetails": [{
      "id": "123",
      "originalPrice": 140.00,
      "displayPrice": 130.00,
      "quantity":2
    }]
  }
}
```

Remove from cart

The following example shows the `remove-from-cart` user event data format.

The `productDetails` section is required, and must include the `quantity` and `id` fields. The `cartId` field is optional.

JAVASCRIPT PIXEL TAG MANAGER

```
 var user_event = {
  "eventType": "remove-from-cart",
  "userInfo": {
    "visitorId": "visitor-id",
    "userId": "user-id"
  },
  "eventDetail": {
    "experimentIds": "experiment-group"
  },
  "productEventDetail": {
    "cartId" : "5678",
    "productDetails": [{
      "id": "123",
      "originalPrice": 140.00,
      "displayPrice": 130.00,
      "quantity": "12"
    }]
  }
}
```

Remove from list

The following example shows the `remove-from-list` user event data format.

The `listId` field is required. The `productDetails` section is required, and must include the `id` field. The `quantity` field is optional.

JAVASCRIPT PIXEL TAG MANAGER

```
 var user_event = {
  "eventType": "remove-from-list",
  "userInfo": {
    "visitorId": "visitor-id",
    "userId": "user-id"
  },
  "eventDetail": {
    "experimentIds": "experiment-group"
  },
  "productEventDetail": {
    "productDetails": [{
      "id": "123",
      "originalPrice": 140.00,
      "displayPrice": 130.00,
      "quantity":2
    }],
    "listId" : "thanksgiving gift ideas"
  }
}
```

Search

The following example shows the `search` user event format.

The `searchQuery` field is required.

JAVASCRIPT PIXEL   TAG MANAGER

```
var user_event = {
   "eventType": "search",
   "userInfo": {
     "visitorId": "visitor-id",
     "userId": "user-id"
     },
   "eventDetail": {
     "experimentIds": "experiment-group"
   },
   "productEventDetail": {
     "searchQuery": "Rainbow Unicorn"
   }
 }
```

Shopping cart page view

The following example shows the `shopping-cart-page-view` user event data format.

The `eventDetails` object is optional. The `productDetails` object is optional, but you should provide it unless the shopping cart is empty. The `cartId` object is optional.

JAVASCRIPT PIXEL TAG MANAGER

```
 var user_event = {
  "eventType": "shopping-cart-page-view",
  "userInfo": {
    "visitorId": "visitor-id",
    "userId": "user-id"
  },
  "eventDetail": {
    "experimentIds": "experiment-group"
  },
  "productEventDetail": {
    "cartId" : "5678",
    "productDetails": [{
      "id": "123",
      "originalPrice": 140.00,
      "displayPrice": 130.00,
      "quantity":2
    },
    {
      "id": "1234",
      "originalPrice": 10.10,
      "displayPrice": 24.50,
      "quantity":1
    }]
  }
}
```

Custom attributes

EventAttributes

You can include additional custom attributes and features for catalog items or user events. This can result in improved, more specific recommendations for your users. To add custom attributes, use the `eventAttributes` field of the `eventDetail` (https://cloud.google.com/recommendations-ai/docs/reference/rest/v1beta1/projects.locations.catalogs.eventStores.userEvents #eventdetail) object when you record a user event.

You can provide custom category values by using the `categoricalFeatures` field, or custom numerical values by using the `numericalFeatures` field.

For example, the following shows the `eventDetail` section from a request to record a user event:

```
"eventDetail":
{
  "eventAttributes":
    {"categoricalFeatures":
      {
        "colors": {"value": ["blue", "red"]},
        "sizes":  {"value": ["XS", "S", "M"] }
      },
      "numericalFeatures":
      {
        "lengths_cm": {"value":[2.3, 15.4] },
        "heights_cm": {"value":[8.1, 6.4] }
      }
    }
}
```

About the user information object

The user information ([`UserInfo`](https://cloud.google.com/recommendations-ai/docs/reference/rest/v1beta1/projects.locations.catalogs.eventStores.userEvents#userinfo)
(https://cloud.google.com/recommendations-ai/docs/reference/rest/v1beta1/projects.locations.catalogs.eventStores.userEvents #userinfo)
) included when you record a user event contains both a `visitorId` and a `userId` value.

`visitorId` represents the unique user identifier and is required when you record a user event. We recommend that you use a secure form of a unique identifier to keep users anonymous to Recommendations AI and protect your users' privacy.

`userId` is optional, and can be used as a unique, persistent identifier for a user across devices whenever a user logs in to your site. When you record the `userId` for a user, Recommendations

AI can generate recommendations based on products that the user browses across multiple devices such as mobile devices and a web browser.

About the timestamp

When you record a user event, be sure to include an accurate timestamp of when the event occurred. Accurate timestamps ensure that Recommendations AI stores events in the correct order. Timestamps are recorded automatically for events collected using Tag Manager and the JavaScript Pixel. When you import events, you must provide the timestamp in the eventTime (https://cloud.google.com/recommendations-ai/docs/reference/rest/v1beta1/projects.locations.catalogs.eventStores.userEvents) field in the format specified by RFC 3339 (https://tools.ietf.org/html/rfc3339).

What's next

- Learn how to record user events (https://cloud.google.com/recommendations-ai/docs/record-events).

Last updated January 9, 2020.