

# Event Notifications Overview

Notifications are payloads of data that are provided as the order progresses through fulfillment stages, which can be sent to a JSON endpoint or custom application. These payloads are triggered automatically when any “event” happens in the fulfillment process - such as when an order is created or changes status. These events may originate from either the user interface or API – notifications are triggered by any appropriate action regardless of where it happens. Because these events occur at the site level, if a tenant-level action occurs (such as the creation of a new customer account) the system triggers the event notification for every site associated with the tenant.

The table in this guide lists some of the most common notifications, identifies the events that trigger them, and links to schemas detailing the parameters that are received in the payloads. These notifications are listed by “topics,” or groups of notifications with related purposes.

## Event APIs

You can use the Event Pull API to retrieve event information. To [retrieve a list of events](#) for a site, run a GET operation using the following URI:

- <http://sandbox.mozu.com/api/event/pull>

To [retrieve more details about a specific event](#) for a site, run a GET operation using the following URI with the event’s unique ID:

- <http://sandbox.mozu.com/api/event/pull/{eventId}>

## Endpoint Configuration

For more information about subscribing custom applications to listen for events through the Event APIs, refer to the [Application Development documentation](#). That documentation includes additional technical guidelines for how the eventing process works and how to configure your endpoint. These configuration best practices are listed below.

### Duplicate Prevention

Kibo expects a 200 OK success response to be returned from within 45 seconds of sending the notification. If no success response is received during that time or if an HTTP status error message is received, the notification will be placed in a queue to be resent every minute or so (the actual timing will vary based on the size of the queue). The notification will continue to be resent until either the success response is received or the notification expires after 14 days.

Due to this process, the endpoint may sometimes receive duplicates of notifications. To avoid saturation with a large number of duplicates, it is recommended to configure the endpoint with duplicate prevention.

### Out-Of-Order Events

It is sometimes possible for notifications of different topics to be received out of order. While Kibo should always send notifications of the same topic in the order of the process flow, notifications from different topic groups are not

guaranteed to be sent in any particular order. For instance, if a shipment event triggers both a Shipment Status and a Shipment Workflow notification, it is not known which will be received first by the endpoint. Thus, it is also advised to configure the endpoint to be capable of receiving out-of-order notifications to avoid any problems that may be caused by expecting these different notifications in a strict order.

## Retry Behavior

It is also important to know that Kibo uses backoff behavior when re-queueing failed notifications. If there is a server outage or other issue preventing notifications from successfully sending, the system will retry several times. The first time the notification fails, there will be a five minute delay before attempting to send it again. The second time it fails, there will be a one hour delay. If it continues to fail, the amount of time between attempts will increase to the point of 24 hours before making the final attempt. If it still cannot go through, the notification will not be tried again.

This means that if orders are placed during a short-term outage that lasts a few hours, the notifications for those orders will eventually come through once the outage is resolved.

## Event Notification Formatting

Notifications will always consist of limited event payload information that you can then use to call the above API and retrieve additional information about that event. To version events, maintain security, and limit payload size, the event payload does not return the object itself, such as an order or product definition - just the IDs.

For instance, when you receive an order.updated notification you can use the order ID in that payload to then make a GET call to the Order API and retrieve the full order data.

## Notification Fields

The body of notification objects contain the following information to describe events.

- **eventID** : A unique identifier for the event.
- **extendedProperties** : Only provided by shipment topic notifications, this is detailed metadata such as the new and old statuses that a shipment moved between. This is a list of key:value string pairs, and the key varies depending on what information is being provided. See the [examples](#) for more information.
- **topic** : The type of event that occurred, such as product.created.
- **entityID** : Identifies the object for which the action occurred, such as a product code or order ID or shipment number.
- **timestamp** : The date-time at which the event occurred.
- **correlationID** : An identifier used by Kibo to track API requests across different services in the data logs.
- **isTest** : Boolean indicator that identifies whether the event was sent as part of a test.

## Notification Topics

Notifications are organized by topics, which are buckets of one or more notifications associated with a particular object such as orders or shipments.

Event Category	Entity ID Represents	Event	Description
Cart	Cart ID	cart.created	The first item was added to a new cart.
		cart.updated	The contents of a defined cart were updated.
		cart.deleted	A cart was deleted. The system deletes a cart when a shopper attempts to retrieve an expired cart, or when an anonymous shopper with an active cart logs in and merges the anonymous cart with the user's cart.
Category	Category ID	category.created	A new product category was created.
		category.updated	The properties of a defined product category were updated. This does not include event notifications when a client adds a product to or removes a product from a category.
		category.deleted	A defined product category was deleted.
Credit	Credit ID	credit.created	A store credit or gift card was created for a customer account.
		credit.updated	The properties of a store credit or gift card were updated. This includes notifications when a client associates the credit with a customer account and notifications when a shopper spends all or part of the credit balance.
		credit.deleted	A store credit or gift card was deleted.
Customer	Customer	customeraccount.created	A customer account was created in the tenant. Because customer accounts are stored at the tenant level, an event notification occurs for each site associated with the tenant.

Event Category	Entity ID Represents	Event	Description
		customeraccount.updated	The properties of a defined customer account were updated.
		customeraccount.deleted	A defined customer account was deleted.
Customer Segment	Customer Segment ID	customersegment.created	A new customer segment was created.
		customersegment.updated	The properties of a defined customer segment were updated. This does not include notifications for when a client adds customer accounts to or removes customer accounts from a customer segment.
		customersegment.customeradded	A customer account was added to the defined customer segment.
		customersegment.customerremoved	A customer account was removed from the defined customer segment.
		customersegment.deleted	A defined customer segment was deleted.
Discount	Discount ID	discount.created	A new discount or coupon was created.
		discount.updated	The properties of a defined discount or coupon were updated. This does not include notifications when a shopper redeems a discount.
		discount.expired	A defined discount or coupon expired.
		discount.deleted	A defined discount or coupon was deleted.
Email	Email ID	email.requested	An email was requested.
Facet	Facet ID	facet.created	A new facet was created.
		facet.updated	A defined facet was updated.
		facet.deleted	A defined facet was deleted.
	Reservation ID	inventory.cartitemallocated	Inventory was successfully allocated for <a href="#">reservation</a> .

Event Inventory Category	Entity ID Represents	Event	Description
	Cart ID	inventory.cartpendingitemcreated	Items were overallocated and a <a href="#">pending item</a> was successfully created.
Location	Location Code	location.created	A new physical location was configured for a tenant environment.
		location.updated	One or more properties of a defined physical location are updated in the tenant environment.
		location.deleted	A defined physical location was deleted from the tenant environment.
	Location Type Code	locationtype.created	A new location type was defined used to group locations.
		locationtype.updated	One or more properties of a location type were updated, or locations were added to or removed from the location type.
		locationtype.deleted	A defined location type was deleted. You cannot delete locations types with associated locations.
Order ID		order.opened	A new order was accepted. This generally occurs after a successful checkout process. However, if you have order validation integrations installed, such as a fraud detection integration, this occurs after the order passes the review process.
		order.fulfilled	The order was shipped to the customer or an in-store pickup was created for fulfillment.
		order.closed	An order was closed after it was fully paid and fulfilled.
		order.cancelled	A defined order was cancelled.

Event Category	Entity ID Represents	Event	Description	
		order.updated	One or more properties of a defined order was updated.	
		order.pendingreview	A defined order is pending further review.	
		order.abandoned	A defined order was abandoned.	
	Return ID	return.cancelled	A defined return was cancelled.	
		return.closed	The return was closed after a return replacement or refund was issued.	
		return.opened	A return was created.	
		return.updated	One or more properties of a defined return are updated.	
		return.rejected	A return is marked as rejected.	
	Payment	Payment ID	payment.refunded	An order payment was refunded.
			producttype.created	A new product type was created in any master catalog. The system triggers an event for each site that features the new product in its associated catalog.
producttype.deleted			A defined product type was deleted.	
producttype.updated			One or more properties of a defined product type were updated.	
product.coderenamed			A defined product code was renamed.	
product.created			A new product was created in any master catalog. The system triggers an event for each site that features the new product in its associated catalog.	
product.updated			One or more properties of a defined product, product bundle, or a variation of a defined product were updated.	
product.deleted			A defined product was deleted.	

Event Category	Entity ID Represents	Event	Description
Product	Product Code	productdraft.created	An update was made to a product definition in draft mode.
		productdraft.deleted	A product definition was deleted in a catalog associated with a master catalog that uses the "Pending" product publishing mode.
		productdraft.discarded	Draft changes to a product definition were discarded.
		productdraft.published	Draft changes to a product definition were published to a live site.
		productdraft.updated	One or more properties of a product definition were updated in an existing draft.
		productinventory.instock	A product previously out of stock returns to a site's active inventory.
		productinventory.outofstock	The active inventory stock of a product was depleted at the specified site.
		productinventory.updated	The number of a product was updated in active inventory stock at the specified site.
		wishlist.created	A new wish list was created and associated with the customer account.
		wishlist.deleted	A wish list associated with the customer account was deleted.
		wishlist.updated	One or more properties of a wish list were updated, or items are added to or removed from a wish list associated with the customer account.
Reservation	Reservation ID	reservation.expired	An <a href="#">inventory reservation</a> expired and the items were deallocated in the cart.

Event Category	Entity ID Represents	Event	Description
Return	Return ID	return.authorized	A return was authorized.
		return.cancelled	A return was cancelled.
		return.closed	A return was processed and closed.
		return.opened	A new return was created.
		return.rejected	A return was rejected instead of authorized.
		return.updated	A return was updated.
Shipment	Shipment ID	shipment.itemAdjusted	A shipment line item was adjusted, such as when the item requirements are changed to make the shipment fulfillable by the assigned location.
		shipment.itemsCancelled	A shipment line item was cancelled.
Shipment Status	Shipment ID	shipment.statuschanged	A shipment changed its status. This is triggered when a shipment moves into any of the following states: READY, REASSIGNED, BACKORDER, CANCELED, FULFILLED, or CUSTOMER_CARE.
Shipment Workflow	Shipment ID	shipment.workflowstatechanged	A shipment changed its workflow state. While this may be related to shipment status events, this is not triggered by the status change itself. Instead, it is triggered by the shipment moving to another step in the fulfillment workflow (such as ACCEPTED_SHIPMENT or PRE_ACCEPT_SHIPMENT, depending on the implementation).
Search	Search Settings	searchsettings.updated	One or more search setting properties were updated.
		site.created	A new site was created.



Event Category	Entity ID Represents	Event	Description
Site	Site ID	site.cloned	A defined site was cloned.
		site.updated	One or more properties of a defined site were updated.
		site.deleted	A defined site was deleted.
Subscriptions	Subscription ID	subscription.statuschanged	A <a href="#">subscription</a> changed status to Active, Paused, Cancelled, or Errored.
		subscription.paymentupdated	A <a href="#">subscription's</a> payment method was updated.
Tenant	Tenant ID	tenant.updated	One or more properties of the defined tenant were updated.
		tenant.deleted	A defined tenant was deleted.