Discounts API Overview

Use the discount-related endpoints of the Catalog Admin and Pricing Storefront API services to configure, manage, and retrieve discount information. This guide provides a reference for when to use each service and links to the API documentation with schema models of each call.

For more information about discounts in the Admin UI or their behavior with orders, see the Discounts user guides.

Catalog Admin APIs

Use the Catalog Admin API service to initially create and manage discounts in your catalog. In addition to endpoints that modify the specific targets and content of existing discounts, Catalog Admin includes these important calls:

- Create Discount
- Get Discount
- Get Discounts
- Update Discount
- Delete Discount
- Get Discount Settings
- Update Discount Settings

It is not recommended to call these GET endpoints from your storefront to retrieve discount information for shoppers. Instead, use the Catalog Admin APIs for configuration of your catalog and then refer to the Pricing Storefront section for the best endpoints to use from your front-end.

> ℹ️ The current version of the Discounts API includes `localizedContent` to support multi-locale catalogs, which was not present in the previous API model. If you were a client prior to May 2024 and have upgraded your implementation to support this feature, you can still access discount data that has not yet been rewritten to the new model by providing an `x-api-version` header set to "1".

Custom Discount Properties

The Update Discount Settings endpoint allows you to create up to a maximum of 10 custom properties for discounts by defining them in the `customProperties` object as shown below. The position determines the order in which properties are displayed in the UI. Those properties will then be available for all discounts within that master catalog.

```json
{
 "stackingConfiguration": {
  "stackingEnabled": true,
  "productOrderLayers": 3,
  "productLineItemLayers": 3,
 },
 "usePostDiscountedOrderValue": false,
 "customProperties": [
  {
   "position": 1,
   "code": "code1",
   "name": "group",
   "enabled": true
  },
  {
   "position": 2,
   "code": "code2",
   "name": "percentoff",
   "enabled": true
  }
 ]
}
```

Once created, use the Create/Update Discount endpoints to set the values on individual discounts by providing the property name and your desired value in the `content.customPropertyValues` and/or `localizedContent.customPropertyValues` objects (depending on your catalog localization).

```json
{
 "id": "discount20",
 "code": "discount20",
 "content":  {
    "localeCode": "en-US",
    "name": "20 Percent Off",
    "friendlyDescription": "This discount gives 20% off your order",
    "customPropertyValues":  {
       "group": "favorites",
       "percentoff": 20
    }
 },
 "localizedContent":  [
  {
    "localeCode": "en-US",
    "name": "20 Percent Off",
    "friendlyDescription": "This discount gives 20% off your order",
    "customPropertyValues":  {
       "group": "favorites",
       "percentoff": 20
    }
  }
 ],
 "amount": 20,
 "amountType": "percentage",
 "status": "active",
 "scope": "Order"

 //continue rest of discount data here
}
```

Pricing Storefront APIs

The Pricing Storefront API service consists of optimized endpoints that allow you to get discount information from the storefront with better performance than using the Catalog Admin or API Extensions.

In addition to specific discounts, you can use these APIs to retrieve a list of all potential discounts that could apply to a given product or products. This gives your storefront more visibility into discounts—for instance, you can use that data to display all applicable discounts in the cart and allow the customer to select which one they want to apply.

Secure vs. Public Access

There is a "public" and a "secure" version of each Pricing Storefront call. When configuring a discount, you will indicate whether the discount is public or not and that will determine which endpoints it can be returned by. This is the `isPublic` field when configuring a discount in the Catalog Admin service.

The public endpoints will only return public discounts, while the secure endpoints will return both public and non-public discounts as long as the API user or application has the Discount Read behavior. This allows you to hide certain discounts by making requests for only those that are public. Both endpoints will only return active discounts.

> ℹ️ You can still reference `isPublic` when calling the Get Discounts API within Catalog Admin. In that case, it should be appended as a filter (such as in `...api/commerce/catalog/admin/discounts/?filter=isPublic EQ false`). Note that this will not restrict results for the user regardless of the Discount Read behavior.

Pricing Storefront Endpoints

The following GET endpoints retrieve discounts according to a ID, coupon code, or specific product identified in the endpoint:

- Get Discount by ID (Public and Secure)
- Get Discount by Discount Code (Public and Secure)
- Get Discount by Coupon Code (Public and Secure)
- Get Item Discounts by Product (Public and Secure)
- Get Discounts by Label (Public and Secure)

These POST endpoints also allow you to retrieve item- and order-level discounts that are filtered to more complex criteria in the request body. They support a list of multiple product codes, as well as whether the specific product(s) are purchase requirements, whether any are target products, customer segments, discount labels, target type, and location codes. Examples are provided in the following sections.

- Get Item Discounts ([Public](#) and [Secure](#))
- Get Order Discounts ([Public](#) and [Secure](#))

Item-Level Example Request

The following call will retrieve item-level discounts from the public endpoint.

POST .../api/commerce/catalog/storefront/discounts/products/

This example searches for product discounts that have Product-AAA as a purchase requirement and target, the "SALE" label, and are able to be applied by a shopper in the Gold-level customer segment at the Dallas location. If `includeUnconditional` were true, then discounts that are applied when there is no condition set on the discount would be included in the results.

```
{
  "isTarget": true,
  "productCodes": [ "Product-AAA" ],
  "isPurchaseRequirement": true,
  "includeUnconditional": false,
  "customerSegmentIds": [ "GOLD" ],
  "labels": [ "SALE" ],
  "targetType": "Product",
  "locationCodes": [ "DALLAS" ]
}
```

A maximum of 50 discounts will be returned in the response, which is a list of applicable discounts and their basic information.

The `conditions` and `target` properties can be used to collect more details about each discount from this response, as they contain most of the same information as the Catalog Admin endpoints. If `includeUnconditional` were true, then discounts that are applied when there is no condition set on the discount would be included in the results.

```
{
 "discounts": [
 {
  "id": 0,
  "code": "0",
  "startDate": "System.String",
  "scope": "LineItem",
  "amount": 0,
  "amountType": "FreeAutoAdd",
  "doesNotApplyToSalePrice": false,
  "doesNotApplyToProductsWithSalePrice": false,
  "isUnconditional": false,
  "isPublic": false,
  "isSubscriptionDiscount": false,
  "content": {
    "name": "System.String",
    "friendlyDescription": "System.String"
  },
  "conditions": {
    "maxRedemptionCount": "System.Int32",
    "requiresCoupon": "System.Boolean",
```

```
    "couponCode": "System.String",
    "requiresAuthenticatedUser": "System.Boolean",
    "includedPaymentWorkflows": [],
    "customerSegments": [
      {
        "id": "System.Int32"
      }
    ],
    "includedCategories": [
      {
        "categoryId": "System.Int32"
      }
    ],
    "excludedCategories": [
      {
        "categoryId": "System.Int32"
      }
    ],
    "includedProducts": [
      {
        "productCode": "System.String"
      }
    ],
    "excludedProducts": [
      {
        "productCode": "System.String"
      }
    ],
    "minimumQuantityRequiredProducts": "System.Int32",
    "minimumQuantityProductsRequiredInCategories": "System.Int32",
    "minimumOrderAmount": "System.Decimal",
    "maximumOrderAmount": "System.Decimal",
    "categoriesToExcludeFromMinOrderTotal": [
      {
        "categoryId": "System.Int32"
      }
    ],
    "productsToExcludeFromMinOrderTotal": [
      {
        "productCode": "System.String"
      }
    ],
    "minimumLifetimeValueAmount": "System.Decimal",
    "minimumCategorySubtotalBeforeDiscounts": "System.Decimal",
    "minimumRequiredQuantityPerRedemption": "System.Int32",
    "validSubscriptionFrequencies": [],
    "validSubscriptionContinuityOrdinals": [],
    "minDistinctProductsRequired": "System.Int32",
    "minTotalOrderQuantity": "System.Int32",
    "pricingContext": "System.String",
    "continuityRecurrenceSettings": {
      "startOrdinal": "System.Int32",
      "interval": "System.Int32",
      "stopOrdinal": "System.Int32"
    },
    "validFulfillmentMethods": [],
    "exclusiveSiteIds": [],
    "exclusiveLocationCodes": []
  },
  "target": {
```

```
  "type": "Product",
  "includeAllProducts": false,
  "includedCategoriesOperator": "Any",
  "excludedCategoriesOperator": "Any",
  "products": [
    {
      "productCode": "System.String"
    }
  ],
  "excludeItemsWithExistingShippingDiscounts": false,
  "excludeItemsWithExistingProductDiscounts": false,
  "maximumQuantityPerRedemption": 1
},
"maximumUsesPerUser": "System.Int32",
"maximumDiscountImpactPerOrder": "System.Decimal",
"maximumDiscountImpactPerRedemption": "System.Decimal",
"maximumRedemptionsPerOrder": "System.Int32",
"includedPriceLists": [],
"doesNotApplyToMultiShipToOrders": "System.Boolean",
"pricingContext": "Any",
"purchaseRequirementType": "Combination",
"isGroupDiscount": false,
"doesNotApplyToMultiShipToOrders": false,
"isBxGx": false,
"canBeStackedUpon": false,
"stackingLayer": 1,
"ThresholdMessage": {
  "discountId": "System.Int32",
  "message": "System.String",
  "thresholdValue": "System.Decimal",
  "showOnCheckout": "System.Boolean",
  "showInCart": "System.Boolean",
  "requiresCouponCode": "System.Boolean",
},
"preventLineItemShippingDiscounts": "System.Boolean",
"preventOrderProductDiscounts": "System.Boolean",
"preventOrderShippingDiscounts": "System.Boolean",
"hasPurchaseConditions": true,
"labels": []
}
```

Order-Level Example Request

The following call will retrieve order-level discounts from the public endpoint.

POST .../api/commerce/catalog/storefront/discounts/orders/

This example searches for shipping discounts that have either Product A or B as a purchase requirement, the "summer" and "clearance" labels, and are able to be applied by a shopper in Customer Segment 123 at two potential locations.

```
{
  "productCodes" : [ "A", "B"],
  "isPurchaseRequirement": true,
  "customerSegments": [123],
  "includeUnconditional": false,
  "labels": ["summer", "clearance"],
  "targetType": "Shipping",
  "locationCodes": ["AUS", "FDW"]
}
]
}
```

A maximum of 50 discounts will be returned in the response, which is a list of applicable discounts and their basic information.

The `conditions` and `target` properties can be used to collect more details about each discount from this response, as they contain most of the same information as the Catalog Admin endpoints. If `includeUnconditional` were true, then discounts that are applied when there is no condition set on the discount would be included in the results.

```
{
 "discounts": [
{
 "id": 0,
 "code": "0",
 "startDate": "System.String",
 "scope": "Order",
 "amount": 0,
 "amountType": "FreeAutoAdd",
 "doesNotApplyToSalePrice": false,
 "doesNotApplyToProductsWithSalePrice": false,
 "isUnconditional": false,
 "isPublic": false,
 "isSubscriptionDiscount": false,
 "content": {
   "name": "System.String",
   "friendlyDescription": "System.String"
 },
 "conditions": {
   "maxRedemptionCount": "System.Int32",
   "requiresCoupon": "System.Boolean",
   "couponCode": "System.String",
   "requiresAuthenticatedUser": "System.Boolean",
   "includedPaymentWorkflows": [],
   "customerSegments": [
     {
       "id": "System.Int32"
     }
   ],
   "includedCategories": [
     {
       "categoryId": "System.Int32"
     }
   ],
   "excludedCategories": [
     {
       "categoryId": "System.Int32"
     }
```

      ],
      "includedProducts": [
        {
          "productCode": "System.String"
        }
      ],
      "excludedProducts": [
        {
          "productCode": "System.String"
        }
      ],
      "minimumQuantityRequiredProducts": "System.Int32",
      "minimumQuantityProductsRequiredInCategories": "System.Int32",
      "minimumOrderAmount": "System.Decimal",
      "maximumOrderAmount": "System.Decimal",
      "categoriesToExcludeFromMinOrderTotal": [
        {
          "categoryId": "System.Int32"
        }
      ],
      "productsToExcludeFromMinOrderTotal": [
        {
          "productCode": "System.String"
        }
      ],
      "minimumLifetimeValueAmount": "System.Decimal",
      "minimumCategorySubtotalBeforeDiscounts": "System.Decimal",
      "minimumRequiredQuantityPerRedemption": "System.Int32",
      "validSubscriptionFrequencies": [],
      "validSubscriptionContinuityOrdinals": [],
      "minDistinctProductsRequired": "System.Int32",
      "minTotalOrderQuantity": "System.Int32",
      "pricingContext": "System.String",
      "continuityRecurrenceSettings": {
        "startOrdinal": "System.Int32",
        "interval": "System.Int32",
        "stopOrdinal": "System.Int32"
      },
      "validFulfillmentMethods": [],
      "exclusiveSiteIds": [],
      "exclusiveLocationCodes": []
    },
    "target": {
      "type": "Shipping",
      "includeAllProducts": false,
      "includedCategoriesOperator": "Any",
      "excludedCategoriesOperator": "Any",
      "products": [
        {
          "productCode": "System.String"
        }
      ],
      "excludeItemsWithExistingShippingDiscounts": false,
      "excludeItemsWithExistingProductDiscounts": false,
      "maximumQuantityPerRedemption": 1
    },
    "maximumUsesPerUser": "System.Int32",
    "maximumDiscountImpactPerOrder": "System.Decimal",
    "maximumDiscountImpactPerRedemption": "System.Decimal",

```
  "maximumRedemptionsPerOrder": "System.Int32",
  "includedPriceLists": [],
  "doesNotApplyToMultiShipToOrders": "System.Boolean",
  "pricingContext": "Any",
  "purchaseRequirementType": "Combination",
  "isGroupDiscount": false,
  "doesNotApplyToMultiShipToOrders": false,
  "isBxGx": false,
  "canBeStackedUpon": false,
  "stackingLayer": 1,
  "ThresholdMessage": {
    "discountId": "System.Int32",
    "message": "System.String",
    "thresholdValue": "System.Decimal",
    "showOnCheckout": "System.Boolean",
    "showInCart": "System.Boolean",
    "requiresCouponCode": "System.Boolean",
  },
  "preventLineItemShippingDiscounts": "System.Boolean",
 "preventOrderProductDiscounts": "System.Boolean",
 "preventOrderShippingDiscounts": "System.Boolean",
 "hasPurchaseConditions": true,
  "labels": []
}
]
}
```