



## PRODUCT MANAGEMENT

# Managing SKUs

Gain a better understanding of a SKU's attributes and how they affect downstream processes

On this page, you'll find information on a [SKUs](#) key attributes.

■ Depending on how you send product data in [create checkout](#) or [create checkout-session](#) requests, your [SKUs](#) might contain [compliance and/or basic product data](#).

Once you define a [SKUs](#) attributes, you can perform a [create](#) or [upsert](#) operation. After that, the resource can be [updated](#) or [deleted](#). You can also [retrieve a SKU by its unique identifier](#) or [search for SKUs using various query parameters](#).

## Physical or digital

A SKU's `physical` attribute indicates whether Digital River has classified the product as physical or digital.

If `physical` is `true`, then the SKU's `taxCode` or the tax code of the SKU's [SKU group](#) represents a physical product, and vice versa.

For details, refer to [how products are classified as physical or digital](#) on the [Product basics](#) page.

## Unique identifier

A [SKU's](#) ■ unique identifier is represented by `id`.

A product's identifier in your system should ideally match the SKU's `id` in ours. This ensures that synchronization is possible and that SKUs work properly throughout an [order's lifecycle](#). Therefore, we recommend that you specify your own `id` and ensure it matches your product's universally unique identifier.

If you don't specify `id` in the body of a [create SKU request](#), Digital River generates and returns a unique value.

In [create](#) or [upsert](#) SKU requests, `id` must adhere to the following character set:

- Digits from `0` to `9`
- Upper case letters from `A` to `Z`
- Lowercase letters from `a` to `z`
- Dots/points: `.`
- Hyphens: `-`
- Underscores: `_`

If `id` contains any other character, then a `400 Bad Request` is returned:

```
{
  "type": "bad_request",
  "errors": [
    {
      "code": "invalid_parameter",
      "parameter": "id",
      "message": "'89&56' is not a valid SKU ID."
    }
  ]
}
```

## Item breadcrumb

A SKU's `itemBreadcrumb` represents a hierarchical classification system that organizes and categorizes your products based on their attributes, characteristics, and relationships. You'll typically structure `itemBreadcrumb` by defining a broad top-level category and, as you move down the hierarchy, getting more specific at each subsequent level.

You might have built similar data structures that inventory your products and make it easier for customers to find what they want. In that case, we recommend using this same data.

If you're engaging the [item classification service](#), then all of your [physical SKUs](#) must have an `itemBreadcrumb`.

■ Even if you're not currently making [cross-border](#) sales, it's best practice to define this parameter so that, in the event you do start selling internationally, you're better positioned to get started.

We recommend making your taxonomies as detailed as possible. This will increase the probability that the [tariff codes](#) returned by this service are accurate.

For example, `Clothing > Women's Jeans` is an acceptable value but `Clothing > Women's Clothing > Jeans > Bootcut Jeans` will likely result in more accurate classifications. You can also define multiple hierarchies, just make sure to separate each with a `;` (semi-colon).

Although not technically required, it's highly recommended that you define `itemBreadcrumb` in English. If its value is in a different language, the classification service will disregard this data point.

## Country of origin

A SKU's `countryOfOrigin` is a two-letter [Alpha-2 country code](#) ■ described in the [ISO 3166](#) ■ international standard. It represents the country where a product was manufactured.

If a SKU does not have a `skuGroupId`, then it must contain a `countryOfOrigin`.

If you submit an invalid country code, then Digital River returns a `400 Bad`

Request :

```
{
  "type": "bad_request",
  "errors": [
    {
      "code": "invalid_parameter",
      "parameter": "countryOfOrigin",
      "message": "'KP' is not a valid Country of Origin."
    }
  ]
}
```

## ECCN



A `SKU's eccn` represents its [Export Control Classification Number](#) (ECCN).

If a SKU does not have a `skuGroupId`, then it must contain an `eccn`.

This value determines whether a product:

- Requires a U.S. export/re-export license
- Contains any other license requirements/restrictions
- Has an end use which is prohibited by applicable export control laws

Digital River's [legal documentation](#) lists ECCNs approved for use in the Digital River APIs. In the table's description field, you may find additional requirements and restrictions that further limit the use of the ECCN.

 Digital River can only resell products with these listed ECCNs. If you have a product with an ECCN that you'd like to be considered for addition to the list, contact [Legal-compliance@digitalriver.com](mailto:Legal-compliance@digitalriver.com) 

## Harmonized system code

A [SKU's](#) `hsCode` represents a [Harmonized System code](#)■.

The format of the code is `####.#.#`, where `#` represents a numeric digit between 0 and 9. The first six digits are mandatory. Any code longer than six digits but less than ten digits is optional and based on the country's preference. The period is not included in the character count.

Digital River validates the format of a Harmonized System code. We don't, however, determine whether `hsCode` accurately classifies your product.

For example, the full ten-digit code for Jasmine rice in the United States is 1006.20.4025. If that's the value you specify, we don't check to ensure your product belongs in that category.

If you submit an incorrectly formatted `hsCode`, then Digital River returns a `400`

Bad Request :

#### Error

```
{
  "type": "bad_request",
  "errors": [
    {
      "code": "invalid_parameter",
      "parameter": "hsCode",
      "message": "'1234.56.YW' is not a valid HS code."
    }
  ]
}
```

## SKU group identifier

The `skuGroupId` uniquely identifies the [SKU group](#) associated with the [SKU](#). For details, refer to [Grouping SKUs](#).

# Manufacturer ID and part number

A [SKU's](#) `manufacturerId` and `partNumber` are only applicable to [physical products](#).

The `manufacturerId` represents the unique identifier of a part/product's manufacturer.

A manufacturer part number (MPN) is a unique code a manufacturer issues that identifies a part/product. This is represented by a SKU's `partNumber`.

MPNs are meant to be static identifiers of a part/product, universal to all distributors, wholesalers, and resellers. They allow customers to identify exact parts and protect themselves from counterfeits accurately.

If two parts/products originate from two manufacturers, each must have its own MPN. These identifiers are especially relevant for automotive and consumer electronics due to the numerous parts in these complex products.

## Managed fulfillments

In [Digital River coordinated fulfillments](#) that use the [orchestrated model](#), you'll need to meet our [managed fulfillment requirements](#) before [creating and updating SKU-inventory item pairs](#).

### Managed fulfillment requirements

All the [checkout's physical SKUs](#) must meet the following requirements:

SKU attribute	Requirement	Sample error
managedFulfillment	The specified value must be DRGlobalFulfillment .	400 Bad Request
		{ "type": "bad_request",
		"errors": [ {
		"code": "invalid_parameter",
taxCode	The specified value must be associated with a <a href="#">physical product</a> .	"parameter": "managedFulfillment", "message": "'DRGlobalFulfillment^
		&%^' is not a valid item type." }
		] }
		400 Bad Request
partNumber	The value must be specified.	{ "type": "bad_request",
		"errors": [ {
		"code": "invalid_parameter",
		"parameter": "partNumber",
		"message": "'null' is not a valid parameter." }
		] }
		400 Bad Request
		{ "type": "bad_request",
		"errors": [ {
		"code": "invalid_parameter",
		"parameter": "partNumber",
		"message": "'null' is not a valid parameter." }
		] }

		409 Conflict
		{ "type": "conflict",
		"errors": [ {
		"code":
		"invalid_parameter",
		"parameter":
manufacturerId	If you specify this value,	"manufacturerId",
	then the partNumber	"message": "Invalid
	must be valid. In other	manufacturer ID. The
	words, the combination of	manufacturer ID and
	manufacturerId and	manufacturer part #
	partNumber must be	combination could not
	correct.	be found." }
		] }

## Creating and updating SKU-Inventory Item pairs

When you [create](#) or [upsert](#) a SKU that meets our [managed fulfillment requirements](#), Digital River synchronously creates an associated [inventory item](#). These [SKU-inventory item pairs](#) have [shared attributes](#) with identical values.



POST /skus

GET /inventory-items/{id}

## Request

```
curl --location --request POST 'https://api.digitalriver.com/skus' \
--header 'Authorization: Bearer <API_key>' \
--header 'Content-Type: application/json' \
...
--data-raw '{
  "id": "sku_demo_1617140947939",
  "eccn": "EAR99",
  "countryOfOrigin": "DE",
  "taxCode": "4323.310_A",
  "partNumber": "DEMOPARTNUMBER2",
  "name": "name_demo_1617140947939",
  "weight": 8.88,
  "weightUnit": "oz",
  "metadata": {
    "application": "iOS-LLL"
  },
  "managedFulfillment": "DRGlobalFulfillment"
}'
```

## Response

```
{
  "id": "sku_demo_1617146803587",
  "createdTime": "2021-03-30T23:26:43Z",
  "name": "name_demo_1617146803587",
  "eccn": "EAR99",
  "partNumber": "DEMOPARTNUMBER2",
  "taxCode": "4323.310_A",
  "countryOfOrigin": "DE",
  "metadata": {
    "application": "iOS-LLL"
  },
  "weight": 8.88,
  "weightUnit": "oz",
  "fulfill": false,
  "allowOversell": true,
  "liveMode": false,
  "managedFulfillment": "DRGlobalFulfillment",
  "physical": true
}
```

Since the SKU's `id` and the inventory item's `id` are the same, you can use this value to [describe items in checkouts](#), [check inventory levels](#), [request shipping quotes](#) and [make reservations](#).

Every subsequent modification of a `managedFulfillment` SKU (assuming the [update request](#) satisfies our [managed fulfillment requirements](#)) synchronously updates the paired inventory item.

POST /skus/{id}

GET /inventory-items/{id}

## Request

```
curl --location --request POST 'https://api.digitalriver.com/skus/sku_demo2_1617153759250' \
--header 'Authorization: Bearer <API_key>' \
--header 'Content-Type: application/json' \
--data-raw '{
  "countryOfOrigin": "US"
}'
```

## Response

```
{
  "id": "sku_demo2_1617153759250",
  "createdTime": "2021-03-31T01:22:39Z",
  "name": "name_demo2_1617153759250",
  "eccn": "EAR99",
  "partNumber": "DEMOPARTNUMBER2",
  "updatedAt": "2021-03-31T01:23:04Z",
  "taxCode": "4323.310_A",
  "countryOfOrigin": "US",
  "metadata": {
    "application": "iOS-LLL"
  },
  "weight": 8.88,
  "weightUnit": "oz",
  "fulfill": false,
  "allowOversell": true,
  "liveMode": false,
  "managedFulfillment": "DRGlobalFulfillment",
  "physical": true
}
```

Once Digital River creates an [inventory item](#), do not directly update or delete it through the [Inventory Items API](#). These operations modify and delete the inventory item but do not affect its paired SKU.

When you want to remove Digital River as a product's fulfillment coordinator, set `managedFulfillment` to `null` in an [update SKU request](#). This deletes the inventory item paired with the SKU. If you then attempt to retrieve the inventory item, you receive a `404 Not Found`.

POST /skus/{id}

GET /inventory-items/{id}

### Request

```
curl --location --request POST 'https://api.digitalriver.com/skus/sku_demo_1617153057380' \
--header 'Authorization: Bearer <API_key>' \
--header 'Content-Type: application/json' \
...
--data-raw '{
  "managedFulfillment": null
}'
```

### Response

```
{
  "id": "sku_demo_1617153057380",
  "createdTime": "2021-03-31T01:10:57Z",
  "name": "name_demo_1617153057380",
  "eccn": "EAR99",
  "partNumber": "DEMOPARTNUMBER2",
  "updatedAt": "2021-03-31T01:19:14Z",
  "taxCode": "4323.310_A",
  "countryOfOrigin": "DE",
  "metadata": {
    "application": "iOS-LLL"
  },
  "weight": 8.88,
  "weightUnit": "oz",
  "fulfill": false,
  "allowOversell": true,
  "liveMode": false,
  "physical": true
}
```


## Deleting SKU-Inventory Item pairs

A [delete SKU request](#) deletes the SKU and the inventory item it's paired with.

## Name and description

A [SKU's](#) `name` should represent the product's brand name and `description` should provide more details about the product.

A `name` is required but `description` is optional. If you assign a string to `description`, we recommend limiting its length to 1000 characters or less.

 In [low-code checkouts](#), `name` is displayed in the order summary section.

## Image and url

You can use a [SKU's](#):

- `image` to specify the URL of a resource that holds the product's image. This resource should be similar to the image(s) you display to customers while they review your store's products.
- `url` to specify the address of a resource that contains the product's description.

If your integration gives customers the option to use [Klarna](#) then each of your SKUs must contain an `image` and `url`.

In [Prebuilt Checkout](#), the `image` of each SKU referenced in the [checkout-sessions'](#) `items[]` is displayed in the order summary section of the [checkout modal](#).

## Weight and weight unit

For [physical SKUs](#), you should provide a product's `weight` denoted in a `weightUnit`. The enumerated `weightUnit` values are `oz`, `lb`, `g`, and `kg`. If you provide `weight` but not a `weightUnit`, then the value defaults to `oz`.

If your site intends to sell physical products [cross-border](#), then we recommend that you pass the `weight` and `weightUnit` of all of your catalog's [physical SKUs](#).

In some countries, like Switzerland, customs officials use a product's weight when calculating import duties. Without this data, Digital River is unable to calculate [landed costs](#).


If you send a [create](#) or [update](#) checkout request whose `shipTo.address.country` represents one of these nations, and any SKU referenced in the [checkout's](#) `items[]` is missing `weight`, then the following error is thrown:

#### 400 Bad Request

```
{
  "type": "bad_request",
  "errors": [
    {
      "code": "missing_parameter",
      "message": "The weight is missing.",
      "parameter": "weight"
    }
  ]
}
```

## Allow oversell

In [Digital River coordinated fulfillments](#), you may build your integration so customers can reserve out-of-stock or pre-order items not yet in stock. To do this, you can use the SKU's `allowOversell` flag.

 In [version](#) 2021-12-13 and later, you can't set a [SKU's](#) `allowOversell`.

If you set the flag to `true`, customers can [reserve the item](#) even when inventory is not available. When set to `false`, customers won't be able to place a hold on an item when its inventory levels are insufficient.

# Tax code

A [SKU's](#) `taxCode` determines whether Digital River classifies a product as [physical or digital](#) and ensures its taxes are properly assessed.

If a SKU does not have a `skuGroupId`, then it must contain a `taxCode`.

You can modify a SKU's `taxCode` in either an [upsert](#) or an [update](#) SKU request.

## Supported tax codes

The following table lists the `taxCode` values supported by Digital River provides a short description of each and indicates whether it represents a [physical or digital product](#).



If you'd like to sell a product that isn't described by any of these `taxCode` (s), contact your account representative.



119KB

Supported tax codes.pdf  
pdf

[Previous](#)  
[Product basics](#)

[Next](#)  
[Using product details](#)

Last updated 2 months ago