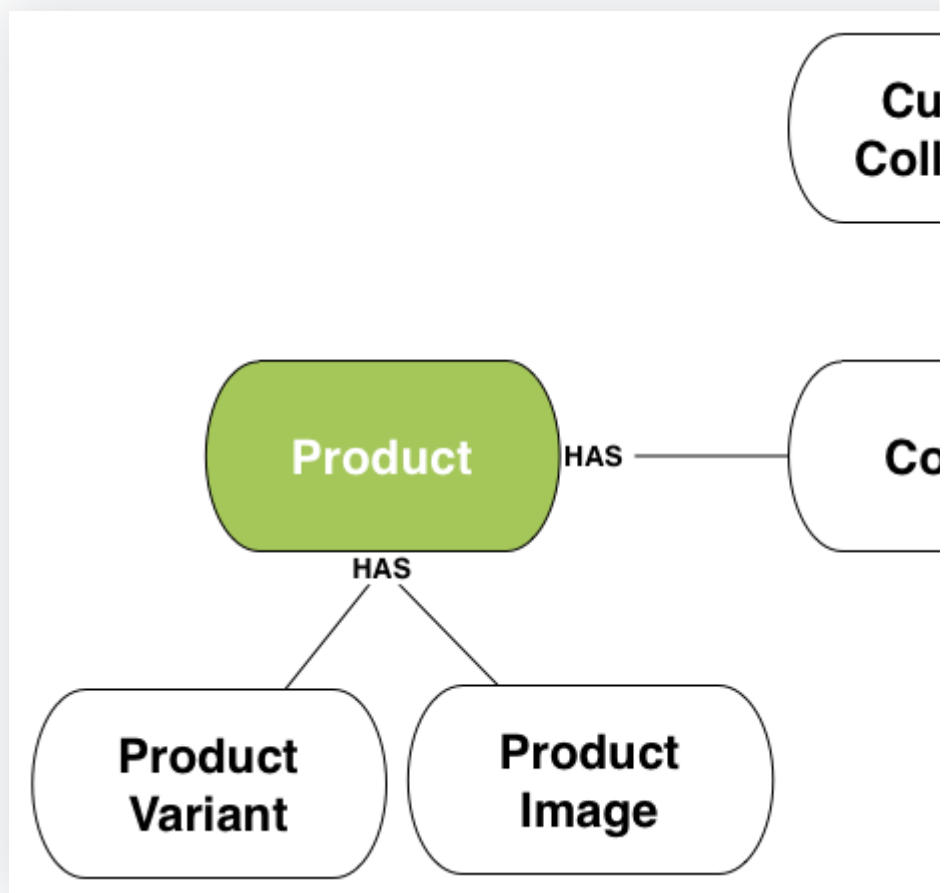


## Product

The Product resource lets you update and create products in a merchant's store. You can use the Product resource to create or update different versions of the same product. You can also add or update product images.

You can add products to collections using the [CustomCollection](#) resource and the [SmartCollection](#) resource.



## What you can do with Product

The Shopify API lets you do the following with the Product resource. More detailed versions of the API are available:

- [GET /admin/products.json](#)

Receive a list of all Products

- [GET /admin/products/count.json?collection\\_id=841564295](#)  
Receive a count of all Products
- [GET /admin/products/{id}.json?fields=id,images,title](#)  
Receive a single Product
- [POST /admin/products.json](#)  
Create a new Product
- [PUT /admin/products/{id}.json](#)  
Modify an existing Product
- [DELETE /admin/products/{id}.json](#)  
Remove a Product from the database

## Product properties

body_html	<code>"body_html": "It's the small iPod with a big idea"</code> Description of the product. Supports HTML formatting.
created_at	<code>"created_at": "2012-02-15T15:12:21-05:00"</code> Date and time when the product was created. The <a href="#">ISO 8601</a> format.
handle	<code>"handle": "ipod-nano"</code> Human-friendly unique string for the product. Auto-generated from the product's <code>title</code> . Used by the Liquid templating language.
id	<code>"id": 632910392</code> Unique numeric identifier for the product. Each <code>id</code> is unique. No two products will have the same <code>id</code> , even if the product is deleted.
images	<code>"images": [{"src": "http://example.com/burton.jpg"}]</code> List of image objects, each one representing an image on the product.
options	<code>"options": [{"name": "Title"}]</code> Custom product property names like <code>Size</code> , <code>Color</code> , etc. Used for permutations of these options. (maximum: 3 options)

product_type	<p><code>"product_type": "Cult Products"</code></p> <p>Categorization that a product can be tagged with, c searching.</p>
published_at	<p><code>"published_at": "2007-12-31T19:00:00-05:00"</code></p> <p>Date and time when the product was published to th returns this value in <a href="#">ISO 8601 format</a>. A value of <code>null</code> published to Online Store.</p>
published_scope	<p><code>"published_scope": "global"</code></p> <p>Indicates whether the product is published to the Po <code>web</code> , <code>global</code> .</p> <ul style="list-style-type: none"> <li><code>web</code> : The product is not published to Point of Sa</li> <li><code>global</code> : The product is published to Point of Sa</li> </ul>
tags	<p><code>"tags": "Emotive, Flash Memory, MP3, Music"</code></p> <p>Categorization that a product can be tagged with. C searching. (maximum: 255 characters for each com</p>
template_suffix	<p><code>"template_suffix": null</code></p> <p>Suffix of the Liquid template being used. By default, <code>product.liquid</code> , without any suffix. Any additional <code>product.suffix.liquid</code> .</p>
title	<p><code>"title": "iPod Nano - 8GB"</code></p> <p>Name of the product. In a shop's catalog, clicking on product's page. On a product's page, the product's t</p>
metafields_global_title_tag	<p><code>"metafields_global_title_tag": "iPod Nano - White"</code></p> <p>Name of the product used for <a href="#">SEO purposes</a>. Genera <code>name='title'&gt;</code> tag.</p>
metafields_global_description_tag	<p><code>"metafields_global_description_tag": "It's the sm"</code></p> <p>Description of the product used for <a href="#">SEO purposes</a>. C <code>name='description'&gt;</code> tag.</p>

updated_at	<div>"updated_at": "2012-08-24T14:01:47-04:00"</div> <p>Date and time when the product was last modified. <a href="#">format</a>.</p>
variants	<div>"variants": [<div>{<div>"barcode": "1234_pink", "compare_at_price": null, "created_at": "2012-08-24T14:01:47-04:00", "fulfillment_service": "manual", "grams": 567, "weight": 0.2, "weight_unit": "kg", "id": 808950810, "inventory_management": "shopify", "inventory_policy": "continue", "inventory_quantity": 10, "option1": "Pink", "position": 1, "price": 199.99, "product_id": 632910392, "requires_shipping": true, "sku": "IPOD2008PINK", "taxable": true, "title": "Pink", "updated_at": "2012-08-24T14:01:47-04:00"</div>}</div>]</div> <p>List of variant objects, each one representing a slight variation of the product. For example, if a product comes in different sizes and colors, the variants attribute would be a permutation (such as <code>small black</code> , <code>medium black</code> , <code>large black</code> ).</p> <p>To reorder variants, update the product with the variant's <code>position</code> attribute on the variant will be ignored.</p> <ul style="list-style-type: none"><li>• <b>barcode</b>: Barcode, UPC, or ISBN number for the product variant.</li><li>• <b>compare_at_price</b>: Competitor's price for the same product variant.</li><li>• <b>created_at</b>: Date and time when the product variant was created. Value in <a href="#">ISO 8601 format</a>.</li><li>• <b>fulfillment_service</b>: Service that is handling fulfillment for the variant, such as <code>gift_card</code> , or the <code>handle</code> of a <a href="#">FulfillmentService</a>.</li><li>• <b>grams</b>: Weight of the product variant in grams.</li></ul>

	<ul style="list-style-type: none"> <li>• <b>weight:</b> Weight of the product variant in the unit</li> <li>• <b>weight_unit:</b> Unit that the product variant's weight is in. <code>kg</code>, <code>oz</code>, <code>lb</code>.</li> <li>• <b>id:</b> Unique numeric identifier for the product variant.</li> <li>• <b>inventory_management:</b> Specifies whether Shopify manages the stock for this product variant.</li> <li>• <b>inventory_policy:</b> Specifies whether customers can purchase the product variant when it's out of stock.</li> <li>• <b>inventory_quantity:</b> Number of items in stock for this product variant.</li> <li>• <b>metafield:</b> Attaches additional information to a product variant.</li> <li>• <b>option:</b> Custom properties that a shop owner can assign to a product variant. Multiple options can exist. Options are represented by <code>option1</code>, <code>option2</code>, <code>option3</code> etc.</li> <li>• <b>position:</b> The order of the product variant in the product's list of variants. To reorder variants, update the product's <code>variant_order</code>. The position attribute on the variant will be updated accordingly.</li> <li>• <b>price:</b> Price of the product variant.</li> <li>• <b>product_id:</b> Unique numeric identifier for the product.</li> <li>• <b>requires_shipping:</b> Specifies whether a customer must provide a shipping address when placing an order for this product variant.</li> <li>• <b>sku:</b> Unique identifier for the product in the shop.</li> <li>• <b>taxable:</b> Specifies whether a tax is charged when the product variant is sold.</li> <li>• <b>title:</b> Title of the product variant.</li> <li>• <b>updated_at:</b> Date and time when the product variant was last updated. Returns this value in <a href="#">ISO 8601 format</a>.</li> </ul>
vendor	<p><b>"vendor": "Apple"</b></p> <p>Name of the vendor of the product.</p>

## Endpoints

GET	/admin/products.json
-----	----------------------

Get a list of products	
ids	Comma-separated list of product ids.
limit	Amount of results. (default: 50) (maximum: 250)
page	Page to show. (default: 1)
since_id	Restrict results to after the specified ID.
title	Filter by product title.
vendor	Filter by product vendor.
handle	Filter by product handle.
product_type	Filter by product type.
collection_id	Filter by collection id.
created_at_min	Show products created after date. (format: 2014-04-25T16:15:47-0
created_at_max	Show products created before date. (format: 2014-04-25T16:15:47
updated_at_min	Show products last updated after date. (format: 2014-04-25T16:15
updated_at_max	Show products last updated before date. (format: 2014-04-25T16:
published_at_min	Show products published after date. (format: 2014-04-25T16:15:4
published_at_max	Show products published before date. (format: 2014-04-25T16:15:
published_status	<ul style="list-style-type: none"> <li>published - Show only published products.</li> <li>unpublished - Show only unpublished products.</li> <li>any - Show all products (default).</li> </ul>
fields	Comma-separated list of fields to include in the response.

## Get all products

GET /admin/products.json

[View Response](#)

## Get all products that belong to a certain collection

GET /admin/products.json?collection\_id=841564295

[View Response](#)

## Get a list of specific products

GET /admin/products.json?ids=632910392,921728736

[View Response](#)

## Get all products after the specified ID

GET /admin/products.json?since\_id=632910392

[View Response](#)

## Get all products, showing only some attributes

GET /admin/products.json?fields=id,images,title

[View Response](#)

GET	/admin/products/count.json
Get a count of all products of a given collection	
vendor	Filter by product vendor.
product_type	Filter by product type.
collection_id	Filter by collection id.
created_at_min	Show products created after date. (format: 2014-04-25T16:15:47-04:00)
created_at_max	Show products created before date. (format: 2014-04-25T16:15:47-04:00)
updated_at_min	Show products last updated after date. (format: 2014-04-25T16:15:47-04:00)

updated_at_max	Show products last updated before date. (format: 2014-04-25T16:
published_at_min	Show products published after date. (format: 2014-04-25T16:15:4
published_at_max	Show products published before date. (format: 2014-04-25T16:15:
published_status	<ul style="list-style-type: none"> <li>published - Show only published products.</li> <li>unpublished - Show only unpublished products.</li> <li>any - Show all products. (default)</li> </ul>

## Counts all products that belong to a certain collection

GET /admin/products/count.json?collection\_id=841564295

[View Response](#)

## Count all products

GET /admin/products/count.json

[View Response](#)

GET

/admin/products/632910392.json

Get a single product

fields

Comma-separated list of fields to include in the response.

## Get only particular fields

GET /admin/products/#{id}.json?fields=id,images,title

[View Response](#)

## Get a single product by ID



GET /admin/products/#{id}.json

[View Response](#)

POST

/admin/products.json

Create a new product

## Create a new product with the default variant and base64 enc

POST /admin/products.json

```
{
  "product": {
    "title": "Burton Custom Freestyle 151",
    "body_html": "<strong>Good snowboard!</strong>",
    "vendor": "Burton",
    "product_type": "Snowboard",
    "images": [
      {
        "attachment": "R0lGODlhAQABAIAAAAAAAAAAACH5BAEAAAAALAAAAABAAEAAAICh"
      }
    ]
  }
}
```

[View Response](#)

## Create a new product with the default product variant

POST /admin/products.json

```
{
  "product": {
    "title": "Burton Custom Freestyle 151",
    "body_html": "<strong>Good snowboard!</strong>",
    "vendor": "Burton",
    "product_type": "Snowboard",
    "tags": "Barnes & Noble, John's Fav, \"Big Air\""
  }
}
```

## Hide Response

HTTP/1.1 201 Created

```
{
  "product": {
    "id": 1071559748,
    "title": "Burton Custom Freestyle 151",
    "body_html": "<strong>Good snowboard!</strong>",
    "vendor": "Burton",
    "product_type": "Snowboard",
    "created_at": "2017-09-22T14:48:44-04:00",
    "handle": "burton-custom-freestyle-151",
    "updated_at": "2017-09-22T14:48:44-04:00",
    "published_at": "2017-09-22T14:48:44-04:00",
    "template_suffix": null,
    "published_scope": "global",
    "tags": "\"Big Air\", Barnes & Noble, John's Fav",
    "variants": [
      {
        "id": 1070325219,
        "product_id": 1071559748,
        "title": "Default Title",
        "price": "0.00",
        "sku": "",
        "position": 1,
        "grams": 0,
        "inventory_policy": "deny",
        "compare_at_price": null,
        "fulfillment_service": "manual",
        "inventory_management": null,
        "option1": "Default Title",
        "option2": null,
        "option3": null,
        "created_at": "2017-09-22T14:48:44-04:00",
        "updated_at": "2017-09-22T14:48:44-04:00",
        "taxable": true,
        "barcode": null,
        "image_id": null,
        "inventory_quantity": 1,
        "weight": 0.0,
        "weight_unit": "lb",
        "old_inventory_quantity": 1,
        "requires_shipping": true
      }
    ],
  },
}
```

```
"options": [  
  {  
    "id": 1022828904,  
    "product_id": 1071559748,  
    "name": "Title",  
    "position": 1,  
    "values": [  
      "Default Title"  
    ]  
  },  
],  
"images": [  
],  
"image": null  
}
```

## Trying to create a product without a title will return an error

POST /admin/products.json

```
{  
  "product": {  
    "body_html": "A mystery!"  
  }  
}
```

[View Response](#)

## Create a new product with the default variant and a product image downloaded by Shopify

POST /admin/products.json

```
{  
  "product": {  
    "title": "Burton Custom Freestyle 151",  
    "body_html": "<strong>Good snowboard!</strong>",  
    "vendor": "Burton",  
    "product_type": "Snowboard",  
    "images": [  
      {  
        "src": "http://example.com/rails_logo.gif"      }  
    ]  
  }  
}
```

```
    }  
  ]  
}  
}
```

[View Response](#)

## Create a new unpublished product

POST /admin/products.json

```
{  
  "product": {  
    "title": "Burton Custom Freestyle 151",  
    "body_html": "<strong>Good snowboard!</strong>",  
    "vendor": "Burton",  
    "product_type": "Snowboard",  
    "published": false  
  }  
}
```

[View Response](#)

## Create a new product with multiple product variants

POST /admin/products.json

```
{  
  "product": {  
    "title": "Burton Custom Freestyle 151",  
    "body_html": "<strong>Good snowboard!</strong>",  
    "vendor": "Burton",  
    "product_type": "Snowboard",  
    "variants": [  
      {  
        "option1": "First",  
        "price": "10.00",  
        "sku": "123"  
      },  
      {  
        "option1": "Second",  
        "price": "20.00",  
        "sku": "123"  
      }  
    ]  
  }  
}
```

```
    ]
  }
}
```

[View Response](#)

## Create a product with a metafield

POST /admin/products.json

```
{
  "product": {
    "title": "Burton Custom Freestyle 151",
    "body_html": "<strong>Good snowboard!</strong>",
    "vendor": "Burton",
    "product_type": "Snowboard",
    "metafields": [
      {
        "key": "new",
        "value": "newvalue",
        "value_type": "string",
        "namespace": "global"
      }
    ]
  }
}
```

[View Response](#)

PUT

/admin/products/632910392.json

Update a product and associated variants and images

## Hide a published product by changing the published attribute

PUT /admin/products/#{id}.json

```
{
  "product": {
    "id": 632910392,
    "published": false
```

```
}  
}
```

[View Response](#)

## Update a product by reordering the product variants

PUT /admin/products/#{id}.json

```
{  
  "product": {  
    "id": 632910392,  
    "variants": [  
      {  
        "id": 457924702  
      },  
      {  
        "id": 39072856  
      },  
      {  
        "id": 49148385  
      },  
      {  
        "id": 808950810  
      }  
    ]  
  }  
}
```

[View Response](#)

## Update a product's title

PUT /admin/products/#{id}.json

```
{  
  "product": {  
    "id": 632910392,  
    "title": "New product title"  
  }  
}
```

[View Response](#)

## Update a product's tags

PUT /admin/products/#{id}.json

```
{
  "product": {
    "id": 632910392,
    "tags": "Barnes & Noble, John's Fav"
  }
}
```

[View Response](#)

## Update a product and one of its variants

PUT /admin/products/#{id}.json

```
{
  "product": {
    "id": 632910392,
    "title": "Updated Product Title",
    "variants": [
      {
        "id": 808950810,
        "price": "2000.00",
        "sku": "Updating the Product SKU"
      },
      {
        "id": 49148385
      },
      {
        "id": 39072856
      },
      {
        "id": 457924702
      }
    ]
  }
}
```

[View Response](#)

## Show a hidden product by changing the published attribute to

```
PUT /admin/products/#{id}.json
{
  "product": {
    "id": 632910392,
    "published": true
  }
}
```

[View Response](#)

---

## Add a metafield to an existing product

```
PUT /admin/products/#{id}.json
{
  "product": {
    "id": 632910392,
    "metafields": [
      {
        "key": "new",
        "value": "newvalue",
        "value_type": "string",
        "namespace": "global"
      }
    ]
  }
}
```

[View Response](#)

---

## Update a product by reordering product image

```
PUT /admin/products/#{id}.json
{
  "product": {
    "id": 632910392,
    "images": [
      {
        "id": 850703190,
        "position": 2
      },
      {
        "id": 562641783,
```



```
      "position": 1
    }
  ]
}
}
```

[View Response](#)

---

## Update a product's SEO title and description

PUT /admin/products/#{id}.json

```
{
  "product": {
    "id": 632910392,
    "metafields_global_title_tag": "Brand new title",
    "metafields_global_description_tag": "Brand new description"
  }
}
```

[View Response](#)

---

## Update a product by adding a new product image

PUT /admin/products/#{id}.json

```
{
  "product": {
    "id": 632910392,
    "images": [
      {
        "id": 850703190
      },
      {
        "id": 562641783
      },
      {
        "src": "http://example.com/rails_logo.gif"
      }
    ]
  }
}
```

[View Response](#)

## Update a product by clearing product images

PUT /admin/products/#{id}.json

```
{
  "product": {
    "id": 632910392,
    "images": [
    ]
  }
}
```

[View Response](#)

DELETE

/admin/products/632910392.json

Delete a product from the shop

## Delete a product along with all its variants and images

DELETE /admin/products/#{id}.json

[View Response](#)

This page was printed on **20 Nov 2017**. For the latest version, please go to <https://help.shopify.com/api/reference/product/>