

SALESmanago

Product API

Updated: November 2022

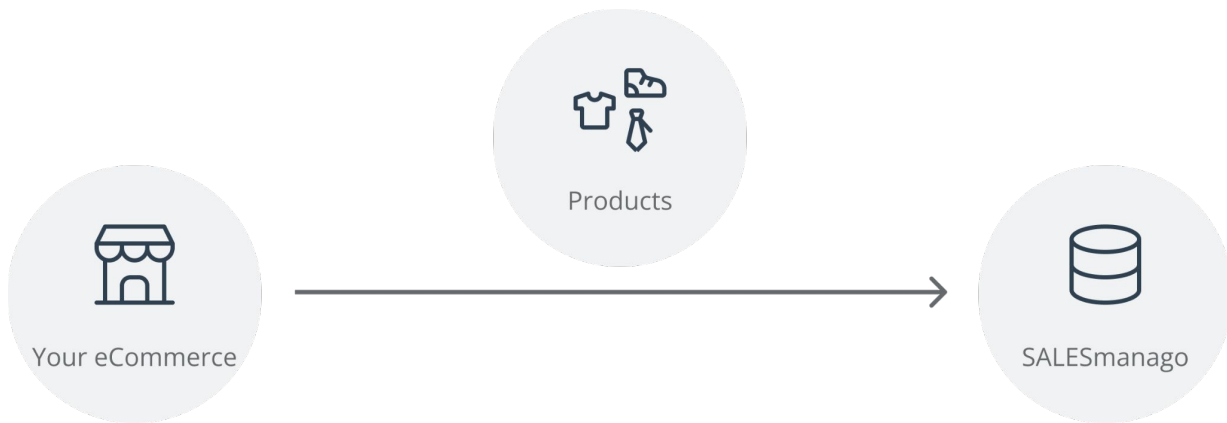
www.salesmanago.com



What is Product API?

Product API is a better alternative to XML Product Feeds. It enables you to send information about the products you offer in your eCommerce store to SALESmanago. This enables you to send product blocks in emails, display products in Recommendation Frames, create product wishlists in your Personal Shopping Inbox, and more.

Product API enables you create **Product Catalogs**.



Product Catalog vs XML Product Feed

	Product Catalog	XML Product Feed
Time to reflect changes	< 30 s	up to 4 h
Performance impact on your store	Small impact every time the product changes	Huge impact every 4 h
Error notifications	Webhook < 120 s	We might contact you by email if the number of errors is high
Fields	Standard product fields Fields available soon: additional product images + category ID + named custom details	Standard product fields
Transfer method	API v3	XML file
Time to market (implementation)	About 4–8 h	About 1 week

How will you benefit from Product API?

Product API is a modern and reliable way of transferring product information to SALESmanago:

- Enables you to display the most up-to-date information in Recommendation Frames, emails, Web Pushes, PSI, and more
- Has much lower impact on the performance of your eCommerce website compared to XML Product Feeds
- Helps you quickly address problems with product updates so all products in SALESmanago are displayed correctly with up-to-date information
- Doesn't require any third-party software to generate XML Product Feed
- Offers easier management of multilanguage or multicurrency stores

What is needed to implement Product API?

To implement Product API on your eCommerce website, you will need:

- API v3 Key created in SALESmanago → Integration Center → API (API v3 tab)
- Access to your website's backend
- Hooks (events) on your eCommerce platform for product creation, update, and deletion

What is API v3?

API is a way for your website (or other services) to communicate with SALESmanago. It's used to transfer and request information from SALESmanago as well as trigger certain actions. In most cases, you will set the API requests to be performed automatically—either periodically or in response to certain events on your website.

API v3 is a new and improved version of the SALESmanago API focusing on real-time performance, better data structures, and error handling. API v3 uses a different authentication method compared to previous API versions—and thus is not compatible with them.

API v3 authentication, part 1

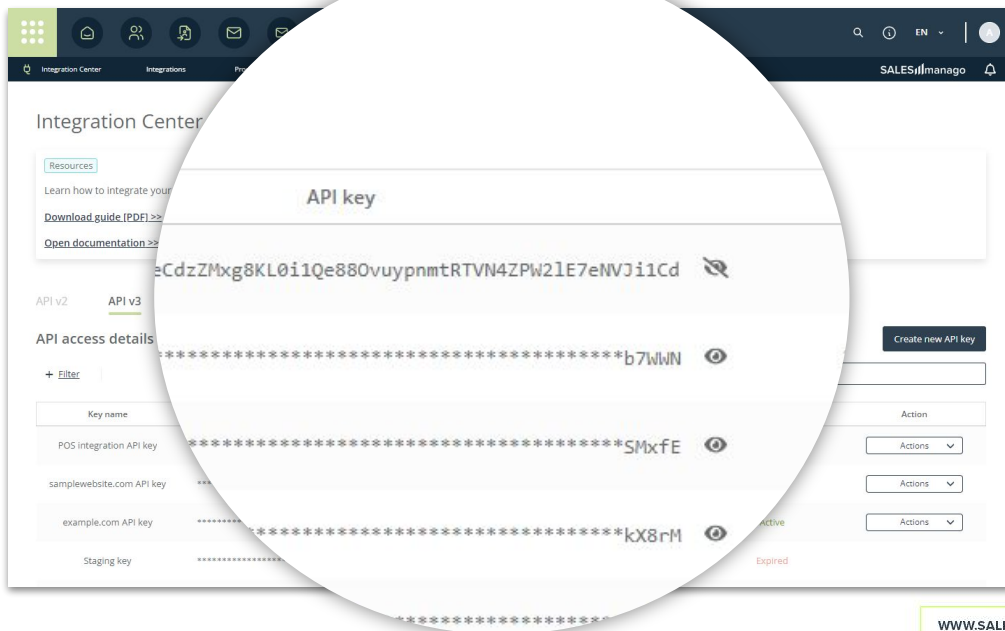
Header name: API-KEY

Header value: <API Key from SALESmanago>

The previous API version used a set of parameters sent in the request body to authenticate as a SALESmanago Client

The new **API v3** uses a simpler, market-standard method—**API-KEY header**

To create your API v3 Key, go to SALESmanago → Integration Center → API (API v3 tab)



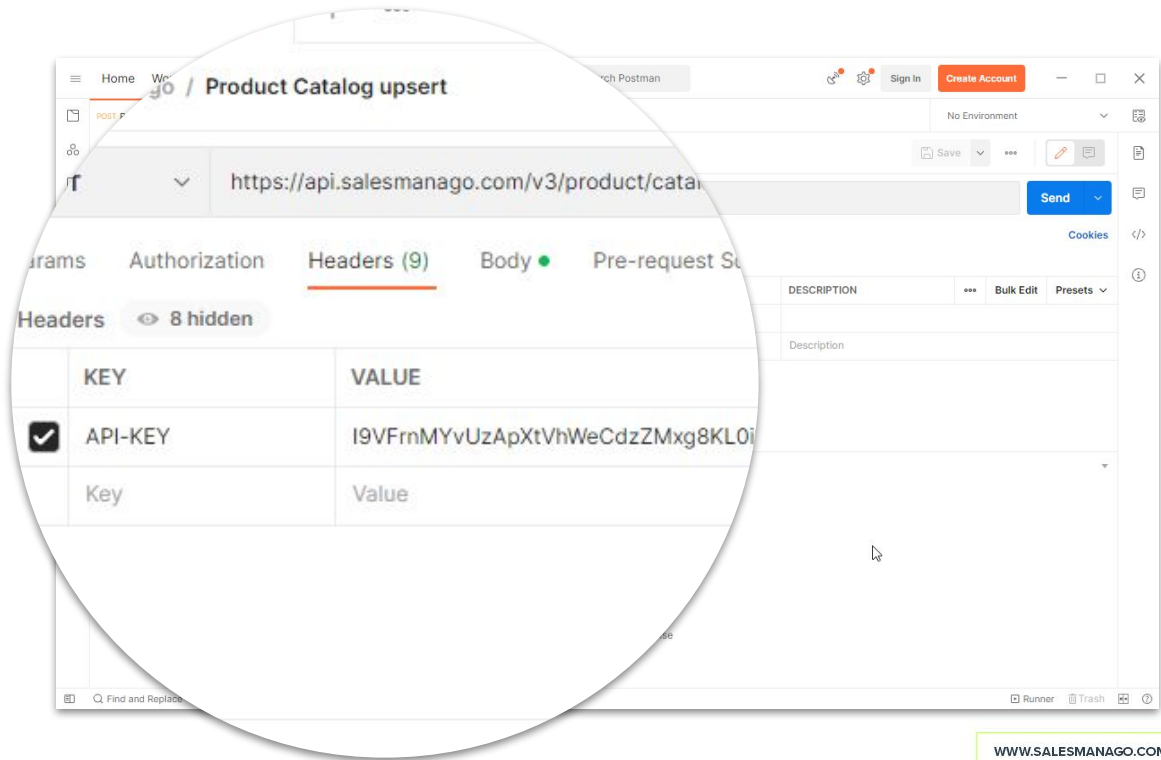
API v3 authentication, part 2

PHP example:

```
<?php
    curl_setopt($ch,
CURLOPT_HTTPHEADER, array(
        'API-KEY:
I9VFrnMYvUzApXtVhWeCdzZMxg8KL0i1Qe8
8OvuypnmtRTVN4ZPW21E7eNVJilCd'
    )
);
?>
```

Let's explore how you would authenticate in Postman:

1. Create a new POST request to the endpoint of a given method
2. Add a new header **API-KEY** in the *Headers* tab
3. Add raw/JSON content in the *Body* tab



Create Product Catalog with API v3

Body example:

```
{
  "catalogName": "Example.com
Product Catalog",
  "setAsDefault": true,
  "currency": "USD",
  "location": "examplecom"
}
```

To create a new Product Catalog, send a POST request to:

`https://api.salesmanago.com/v3/product/catalogUpsert`

(The API v3 uses only one endpoint prefix, i.e., `api.salesmanago.com`)

The body should be `Content-Type: application/json` containing:

Field	Limit	Description
catalogName*	64	Internal name of the Product Catalog to help you identify it in SALESmanago
setAsDefault	bool	Flag used to determine whether Product Catalog should be available for Recommendation Frames
currency*	<u>ISO 4217</u>	ISO currency code, e.g., USD, EUR, etc.
location*	255	Legacy field for assigning products to External Events

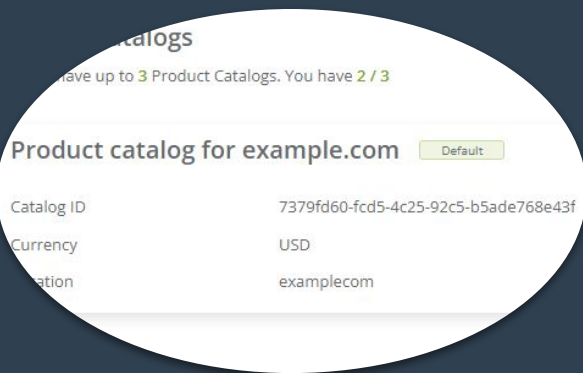
Update Product Catalog with API v3

Body example:

```
{
  "catalogId": "21c2-...-3266",
  "currency": "EUR"
}
```

To update the Product Catalog, add catalogId to the request. You can get the catalogId from the response of the previously described request or from SALESmanago → Integration Center → Product Catalogs.

Field	Limit	Description
catalogName*	64	Internal name of the Product Catalog to help you identify it in SALESmanago.
catalogId	36	Catalog identifier that is used to update an existing catalog. Do not specify when creating a new one.
setAsDefault	bool	Flag used to determine if Product Catalog should be available for Recommendation Frames
currency*	ISO 4217	ISO currency code, e.g. USD, EUR, etc.



Add or update a product in Product Catalog, part 1

Body example:

```
{
  "catalogId": "21c2-...-3266",
  "products": [
    {
      "productId": "1234",
      "name": "Turtleneck sweater",
      "mainCategory": "sweaters",
      "categoryExternalId": "8642",
      "description": "Classic shirt",
      "productUrl":
"https://example.com/shop/shirts/classic",
      "mainImageUrl":
"https://example.com/media/turtleneck-sweater-cover.jpg",
      "available": true,
      "price": 79.99,
      "systemDetails": {
        "manufacturer": "My Brand",
        "gender": 0,
        "color": "black"
      }
    }
  ]
}
```

To add or update a product, send a POST request to:

`https://api.salesmanago.com/v3/product/upsert`

The body should have the `Content-Type: application/json` containing:

Field	Limit	Description
catalogId*	36	Catalog identifier that is used to identify the Product Catalog for the upserted product
products*	100	Array of product objects with fields specified below
productId*	255	Product identifier from your eCommerce platform
name*	255	Product name. You can use diacritics and special characters.
mainCategory*	255	Category name that can be used in emails and when displaying Recommendation Frames
categoryExternalId	255	Category ID used for AI processing and calculating recommendations. If you don't specify this field, the category ID will be assigned based on the mainCategory field.

Add or update a product in Product Catalog, part 2

Body example:

```
{
  "catalogId": "21c2-...-3266",
  "products": [
    {
      "productId": "1234",
      "name": "Turtleneck sweater",
      "mainCategory": "sweaters",
      "categoryExternalId": "8642",
      "description": "Classic shirt",
      "productUrl":
"https://example.com/shop/shirts/classic",
      "mainImageUrl":
"https://example.com/media/turtleneck-sweater-cover.jpg",
      "available": true,
      "price": 79.99,
      "systemDetails": {
        "manufacturer": "My Brand",
        "gender": 0,
        "color": "black"
      }
    }
  ]
}
```

Field	Limit	Description
description	16384	Product description used in emails and custom Recommendation Frames. Most functionalities can use only the first 1024 characters.
productUrl*	2048	Product URL that matches visits with products. Most functionalities can use only the first 512 characters.
mainImageUrl	2048	Image URL to be used in Recommendation Frames and emails. Most functionalities can use only the first 512 characters.
available*	bool	Marks products as temporarily unavailable. This allows you to prevent them from showing up in Recommendation Frames.
active	bool	Marks products as no longer present in your store. This effectively turns off a given product.
quantity	int	Available quantity to be used in custom Recommendation Frames
price*	float (19.2)	Standard product price

Add or update a product in Product Catalog, part 3

Body example:

```
{
  "catalogId": "21c2-...-3266",
  "products": [
    {
      "productId": "1234",
      "name": "Turtleneck sweater",
      (...)
    },
    {
      "productId": "5678",
      "name": "Classic shirt",
      (...)
    },
    {
      "productId": "9012",
      "name": "Casual polo",
      (...)
    }
  ]
}
```

Product upsert lets you add/update up to **100** products with one request:

- You can simultaneously add new and update existing products with a single request
- Some products may contain more details than others
- These mechanics are very useful for the initial transfer of products from your eCommerce store

Field	Limit	Description
products*	100	Array of product objects with the fields specified below

Advanced fields

Body example:

```
(...)  
"systemDetails": {  
  "brand": "My Brand",  
  "manufacturer": "My Brand",  
  "popularity": 40,  
  "gender": 0,  
  "season": "summer",  
  "color": "white",  
  "bestseller": true,  
  "newProduct": false  
},  
"categories": [  
  "Winter collection",  
  "Linen products"  
],  
"imageUrls": [  
  "https://example.com/media/white-t-shirt1.jpg",  
  "https://example.com/media/white-t-shirt2.jpg",  
  "https://example.com/media/white-t-shirt3.jpg"  
],  
"discountPrice": 19.99,  
"customDetails": {  
  "detail1": "linen",  
  "detail2": "short"  
}  
(...)
```

You can send advanced fields for further customization and use in AI Search engine, Voice Search, etc.

Field		Limit	Description
systemDetails		n/a	Set of details that can be used in various mechanisms in SALESmanago
	manufacturer	255	Standard product details
	season	255	Standard product details
	color	25	Standard product details
	popularity	int	Integer value to mark how popular the upserted product is, for example, using a range of 1-100.
	bestseller	bool	Flags that you can use in emails and custom Recommendation Frames
	newProduct	bool	Flags that you can use in emails and custom Recommendation Frames
	gender	enum	Enum to identify the gender the product is designed for: -1 – undefined, 0 – female, 1 – male, 2 – other, 4 – unisex
categories		5*255	Other categories (array) used for recommendations
imageUrls		5*2048	Additional product images that will be used in upcoming features
discountPrice		float (19.2)	Discount product price. You can use this value to display product promo price next to a crossed out standard product price in emails and Recommendation Frames.
customDetails (object)		5*255	List of additional details as key–value pairs. Note: You can use custom key names (instead of detailX), however, some functionalities will still refer to those values using a detailX key.



We are here to **HELP YOU**

If you have any trouble integrating with SALESmanago, contact
Customer Success at success@salesmanago.com.

We will be glad to **guide you directly** through the integration
process!

WWW.SALESMANAGO.COM

