

Tracking products

A product is a good or service offered to visitors on a site.

- [What is product tracking?](#)
- [Tracking products with the data layer](#)
 - [Triggering product view events](#)
 - [Product object](#)
 - [Product object examples](#)
 - [Retail](#)
 - [iGaming](#)
 - [Testing product tracking](#)
- [Checking that the product data is in the Frosmo back end](#)

What is product tracking?

Product tracking is the process of:

- Collecting product data from a site.
- Counting the number of views different products receive from visitors.

Product data and product view statistics are stored in the Frosmo back end.

Product tracking is essential for building a database of products to which [transactions](#) and [other product conversions](#) can be mapped, and for tracking which products visitors view. Product data and product view statistics are a prerequisite for implementing [product recommendations](#) and other features that revolve around products, such as segmenting visitors based on the types of products they have viewed.

You collect product data by either scraping the data from product pages as they load in a visitor's browser or by reading the data from a product feed.

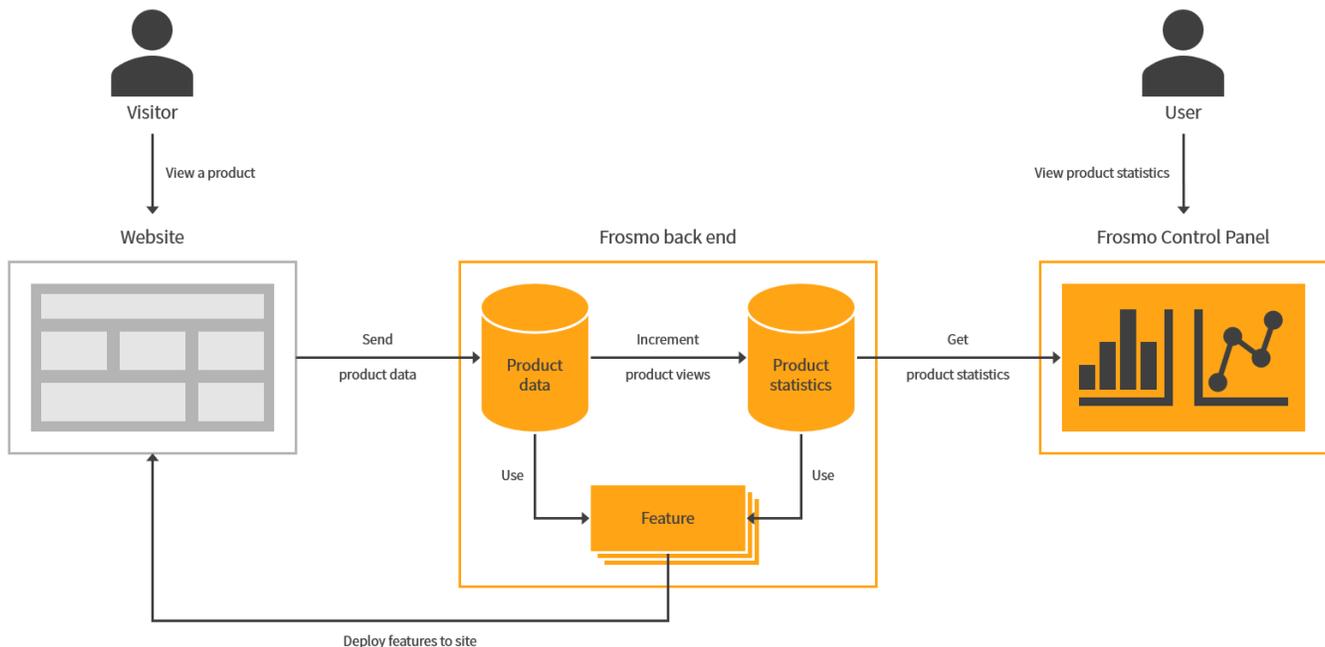


Figure: Product tracking in the Frosmo Platform (click to enlarge)

Product tracking generates product information and statistics, which you can [view in the Frosmo Control Panel](#).



Product tracking does not cover product purchases or any other product-related conversions. If you want to track product purchases or non-purchase product conversions, use [transaction tracking](#) or [conversion tracking](#).

Tracking products with the data layer

Tracking products with the [data layer](#) means triggering a **product view event** whenever a visitor navigates to a product page or otherwise views a product on a site. The data you pass in the product view event is the product data for the viewed product.



To use the data layer on a site, the data layer module must be [enabled for the site](#).

You can trigger product view events from:

- [Modifications](#) (either from [custom content](#) or, if you're using a template, from the [template content](#))
- [Shared code](#)
- Page code (meaning directly from your site source code)

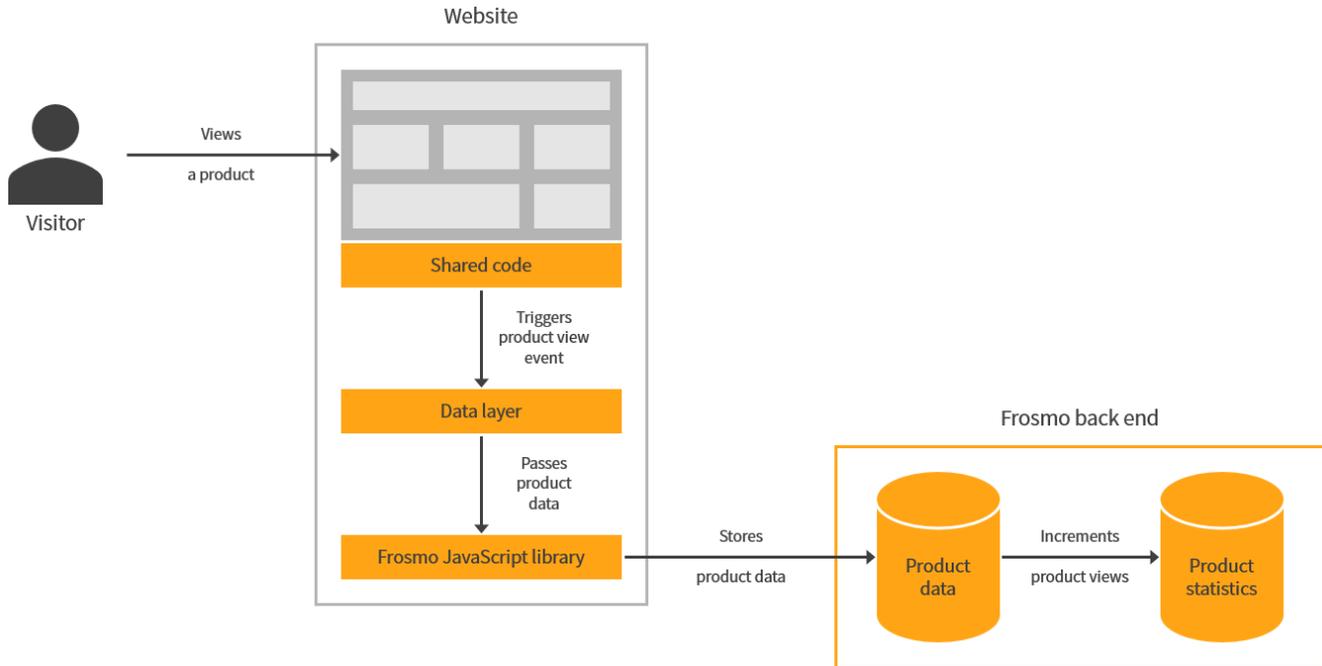


Figure: Tracking products by triggering a product view event from shared code (click to enlarge)

When you trigger a product view event:

- If the product is not in the product database, the Frosmo Platform adds the event data to the database as a new product. The Frosmo Platform uses the product ID to uniquely identify the product.
- If the product is already in the product database, and if the event data differs from the product data in the database, the Frosmo Platform updates the data in the database. The platform only updates product attributes whose values differ and adds attributes that do not already exist in the database. The platform does not remove existing product attributes that are missing from the event data.
- If the product is already in the product database, and if the event data does not differ from the product data in the database, the Frosmo Platform ignores the event data.
- By default, the Frosmo Platform increments the view count for the product by one. For a newly added product, this is the first view for the product.

To check that the product data is correctly stored in the Frosmo back end, [call the Product API](#).

Triggering product view events

To trigger a product view event, call the `dataLayer.push()` function with a **product object** containing the product data:

```
dataLayer.push({
  frosmoProductId: 'string',
  /* Optional */
  frosmoProductCategory: 'string',
  frosmoProductData: {},
  frosmoProductDiscountPrice: 0,
  frosmoProductImage: 'string',
  frosmoProductName: 'string',
  frosmoProductPrice: 0,
  frosmoProductPromotionLabel: 'string',
  frosmoProductUrl: 'string'
  /* ... */
});
```

Product object

The product object contains the product data for a product.

Table: Product object properties

Property	Description	Type	Role
frosmoProductId	<p>Product ID.</p> <p>The Frosmo Platform uses the product ID to uniquely identify the product.</p>	String (256)	Required
frosmoProductArea	<p>Geographical region, typically a part of a country, that applies to the product in some way.</p> <p>For example, if the product is a hotel room, this property can provide the name of the in-country region in which the hotel is located:</p> <ul style="list-style-type: none"> frosmoProductCountry: Greece frosmoProductArea: Rhodes frosmoProductCity: Lindos 	String (32)	Optional
frosmoProductCampaign	Sales or other campaign with which the product is associated.	String (32)	Optional
frosmoProductCategories	<p>Additional product categories to which the product belongs.</p> <p>For more information about the object type, see the instructions below this table.</p>	One of: <ul style="list-style-type: none"> Array of strings Stringified array of strings (1024) Object Stringified object (1024) 	Optional
frosmoProductCategory	<p>Product category or type.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Even if you do not have a value for this property, it is recommended that you nonetheless pass an empty string. This property can be used in segmentation rules and therefore needs to have some value. </div>	String (128)	Optional
frosmoProductCity	<p>City or town that applies to the product in some way.</p> <p>For example, if the product is a hotel room, this property provides the name of the city or town in which the hotel is located:</p> <ul style="list-style-type: none"> frosmoProductCountry: Greece frosmoProductArea: Rhodes frosmoProductCity: Lindos 	String (32)	Optional
frosmoProductColors	<p>Color options for the product.</p> <p>For more information about the object type, see the instructions below this table.</p>	One of: <ul style="list-style-type: none"> Object Stringified object (1024) 	Optional
frosmoProductCompany	Company that offers the product.	String (32)	Optional
frosmoProductConversion	<p>If you want to merely send the product data to the Frosmo back end without triggering a product view event, set this property to 1. In this case, the Frosmo Platform does not increment the view count for the product, but instead just stores the product data.</p> <p>The default value is 0, which means that the Frosmo Platform also registers the product data push as a product view.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  Despite the name, this property does not trigger a conversion event for the product, regardless of the property value. If you want to register a conversion for the product, trigger a conversion event or transaction event instead. </div>	Number	Optional

frosmoProductCountry	<p>Country that applies to the product in some way.</p> <p>For example, if the product is a hotel room, this property provides the name of the country in which the hotel is located:</p> <ul style="list-style-type: none"> • <code>frosmoProductCountry</code>: Greece • <code>frosmoProductArea</code>: Rhodes • <code>frosmoProductCity</code>: Lindos 	String (32)	Optional
frosmoProductCustomImages	<p>URLs of additional product images.</p> <p>For more information about the object type, see the instructions below this table.</p>	<p>One of:</p> <ul style="list-style-type: none"> • Object • Stringified object (2048) 	Optional
frosmoProductData	<p>Any piece of product data that does not logically belong to the other product properties but that you want to push with the product data. Use this property to send your own custom product attributes along with the product data.</p> <p>For more information about the object type, see the instructions below this table.</p>	<p>One of:</p> <ul style="list-style-type: none"> • Object • Stringified object (2048) 	Optional
frosmoProductDescription	Product description.	String (512)	Optional
frosmoProductDiscountPrice	Discount unit price for the product.	String (32)	Optional
frosmoProductIds	<p>Additional product IDs.</p> <p>For example, if the product consists of multiple child products, you can add their IDs to this property.</p> <p>For more information about the object type, see the instructions below this table.</p>	<p>One of:</p> <ul style="list-style-type: none"> • Object • Stringified object (1024) 	Optional
frosmoProductImage	<p>Product image URL.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;">  Even if you do not have a value for this property, it is recommended that you nonetheless pass an empty string. </div>	String (512)	Optional
frosmoProductMainCategory	Main product category to which the product belongs.	String (128)	Optional
frosmoProductManufacturer	Company that produces the product, or the brand name for the product.	String (512)	Optional
frosmoProductModel	Product model name.	String (32)	Optional
frosmoProductMultiPrices	<p>Additional unit price options for the product.</p> <p>For more information about the object type, see the instructions below this table.</p>	<p>One of:</p> <ul style="list-style-type: none"> • Object • Stringified object (2048) 	Optional
frosmoProductName	<p>Product name.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;">  Even if you do not have a value for this property, it is recommended that you nonetheless pass an empty string. </div>	String (128)	Optional
frosmoProductPrice	<p>Unit price for the product.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;">  Even if you do not have a value for this property, it is recommended that you nonetheless pass an empty string. </div>	String (32)	Optional
frosmoProductPromotionLabel	<p>Promotion-related label, such as "Weekend Sale" or "80% off!".</p> <div style="border: 1px solid #ffc107; border-radius: 10px; padding: 10px; margin-top: 10px;">  This property is intended for promotion-related label strings. Do not use this property for stringified objects. If you need to store extra product data beyond the standard product data properties listed here, use the <code>frosmoProductData</code> property. </div>	String (512)	Optional

frosmoProductRating	Product rating.	String (512)	Optional
frosmoProductStock	Number of product units in stock.	Number	Optional
frosmoProductSubCategory	Subcategory to which the product belongs. The subcategory should be a child category of the main product category. For example, if the main product category is "Children", the subcategory could be "Clothes" or "Toys".	String (128)	Optional
frosmoProductUrl	Product page URL. <div style="border: 1px solid #ccc; border-radius: 5px; padding: 5px; margin-top: 10px;">  Even if you do not have a value for this property, it is recommended that you nonetheless pass an empty string. </div>	String (512)	Optional
frosmoProductUrn	Uniform Resource Names (URN) for the product.	String (32)	Optional
frosmoProductVenue	Venue in which the product is available. For example, if the product is a specific showing of a movie, this property would be the movie theater in which the movie is shown.	String (64)	Optional



Objects

Follow these rules for product object properties whose type is object:

- The value can be either a JavaScript object or a [JSON-stringified object](#). In the latter case, the maximum length of the string is either 1024 or 2048 characters, depending on the property (see above).
- The properties of the object can be anything you want them to be. The Frosmo Platform does not validate the properties against any data model or schema.

If you assign a JavaScript object to a product object property, the Frosmo Platform stringifies the value. If you assign a stringified object to a property, the platform stores the value as-is.

Product object examples

Retail

Example: Triggering a product view event with a simple product object

```
dataLayer.push({
  frosmoProductId: '123456',
  frosmoProductCategory: 'Plushies',
  frosmoProductDiscountPrice: 9.99,
  frosmoProductImage: '/images/cheetah_plushy.png',
  frosmoProductName: 'Cheetah Plushy',
  frosmoProductPrice: 19.99,
  frosmoProductPromotionLabel: 'Weekend Sale',
  frosmoProductUrl: '/products/123456'
});
```

Example: Triggering a product view event with a product object containing an object property

```
/* frosmoProductColors is a JavaScript object */

dataLayer.push({
  frosmoProductId: '123456',
  frosmoProductCategory: 'Plushies',
  frosmoProductDiscountPrice: 9.99,
  frosmoProductImage: '/images/cheetah_plushy.png',
  frosmoProductName: 'Cheetah Plushy',
  frosmoProductPrice: 19.99,
  frosmoProductPromotionLabel: 'Weekend Sale',
  frosmoProductUrl: '/products/123456',
  frosmoProductColors: {
    black: '#000000',
    blue: '#0000ff',
    lime: '#00ff00',
    red: '#ff0000'
  }
});

/* frosmoProductColors is a JSON-stringified object */

dataLayer.push({
  frosmoProductId: '123456',
  frosmoProductCategory: 'Plushies',
  frosmoProductDiscountPrice: 9.99,
  frosmoProductImage: '/images/cheetah_plushy.png',
  frosmoProductName: 'Cheetah Plushy',
  frosmoProductPrice: 19.99,
  frosmoProductPromotionLabel: 'Weekend Sale',
  frosmoProductUrl: '/products/123456',
  frosmoProductColors: '{"black":"#000000","blue":"#0000ff","lime":"#00ff00","red":"#ff0000"}'
});
```

Example: Triggering a product view event with minimal data and without incrementing the view count for the product

```
dataLayer.push({
  frosmoProductId: '123456',
  frosmoProductConversion: 1
});
```

Example: Triggering a product view event with multiple product categories for recommendation filtering

```
dataLayer.push({
  frosmoProductId: '234567',
  frosmoProductCategory: '',
  frosmoProductCategories: [
    'Accessories',
    'Bedding',
    'Toys'
  ],
  frosmoProductImage: '/images/cheetah_pillow.png',
  frosmoProductName: 'Cheetah Pillow',
  frosmoProductPrice: 29.99,
  frosmoProductUrl: '/products/234567'
});
```

Example: Triggering a product view event where the product is a bet on a sporting event

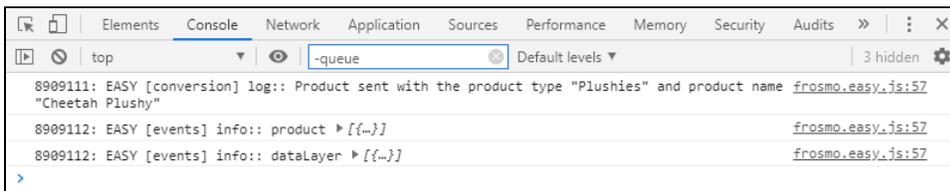
```
dataLayer.push({
  frosmoProductId: 'bet-1234567', // Bet Id
  frosmoProductCategory: 'Football / Germany / Bundesliga', // Combination of main event information
  frosmoProductCountry: 'Germany',
  frosmoProductData: {
    date: '2020-06-12 21:30:00'
  },
  frosmoProductMainCategory: 'Football', // Event category (here, sport name)
  frosmoProductModel: 'Live', // Event type
  frosmoProductName: 'TSG Hoffenheim - RB Leipzig', // Event name
  frosmoProductUrl: '/event/1006388033', // Event URL
  frosmoProductVenue: 'Bundesliga' // Event venue (here, sports league)
});
```

Testing product tracking

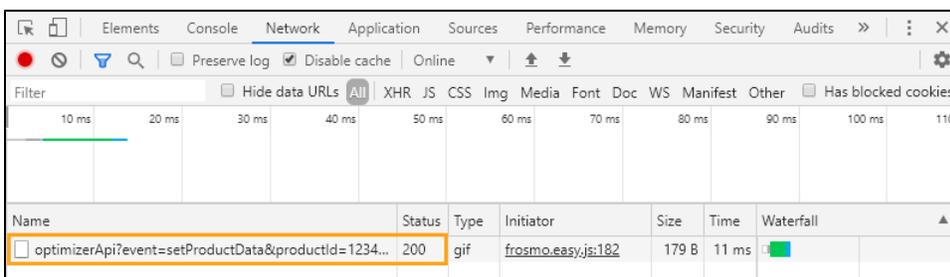
To test that products are correctly tracked with the data layer:

1. Go to the site.
2. [Enable console logging for Frosmo Core.](#)
3. Go to a page where products are tracked. If product view events are successfully triggered with the data layer, the browser console displays the following messages for each event:
 - EASY [conversion] log:: Product sent with the product type "<frosmoProductCategory>" and product name "<frosmoProductName>"
 - EASY [events] info:: product (contains the product data parsed from the product object)
 - EASY [events] info:: dataLayer (contains the product object passed to the data layer)

The following figure shows an example of the product view event messages.



4. If you want more details on a data layer call, select the **Network** tab in the developer tools, and check the `setProductData` request to the Optimizer API. If the status is 200, the request completed successfully.



Checking that the product data is in the Frosmo back end

If you want to verify that the product data you're tracking is correctly stored in the Frosmo back end, try retrieving the data with the [Product API](#).



The Product API omits the `frosmoProduct` prefix from the product property names, and nests all product data under the `attributes` property, except for `id`, `name`, and `type`, which are returned in the product object root.

The `frosmoProductCategory` property in the data layer is returned as the `type` property by the Product API.

Example: Retrieving the product data with the Product API

```
# Request the data for product ID 123456

GET https://example.frosmo.com/productApi?method=fetch&
    origin=shop_company_com&
    ids=["123456"]

# Response for product ID 123456, if the product exists

{
  "data": [
    {
      "id": "123456",
      "type": "Plushies",
      "name": "Cheetah Plushy",
      "created_at": "2020-05-07T14:31:59Z",
      "updated_at": "2020-05-13T15:53:34Z",
      "attributes": {
        "discountPrice": "9.99",
        "image": "/images/cheetah_plushy.png",
        "price": "19.99",
        "promotionLabel": "Weekend Sale",
        "url": "/products/123456"
      }
    }
  ]
}
```