

Tracking transactions

A transaction is the simultaneous purchase of one or more products. The Frosmo Platform registers each transaction, irrespective of the number of products purchased, as a single [conversion](#).

- [What is transaction tracking?](#)
- [Tracking transactions with the data layer](#)
 - [Triggering transaction events](#)
 - [Transaction object](#)
 - [Transaction examples](#)
 - [Testing transaction tracking](#)

What is transaction tracking?

Transaction tracking is the process of monitoring visitors for actions that qualify as transactions and collecting the data about those actions (transaction data). Transaction tracking also involves counting transactions as conversions and attributing those conversions to modifications, which the Frosmo Platform does automatically when it receives transaction data from a site. The data is stored in the Frosmo back end.

Transaction tracking allows you to monitor the revenue generated by your site and measure Frosmo's impact on that revenue. Transaction tracking is also a prerequisite for implementing features that rely on transaction data, such as generating [product recommendations](#) and [segmenting visitors](#) based on the products they have purchased, which in turn feed into revenue generation.

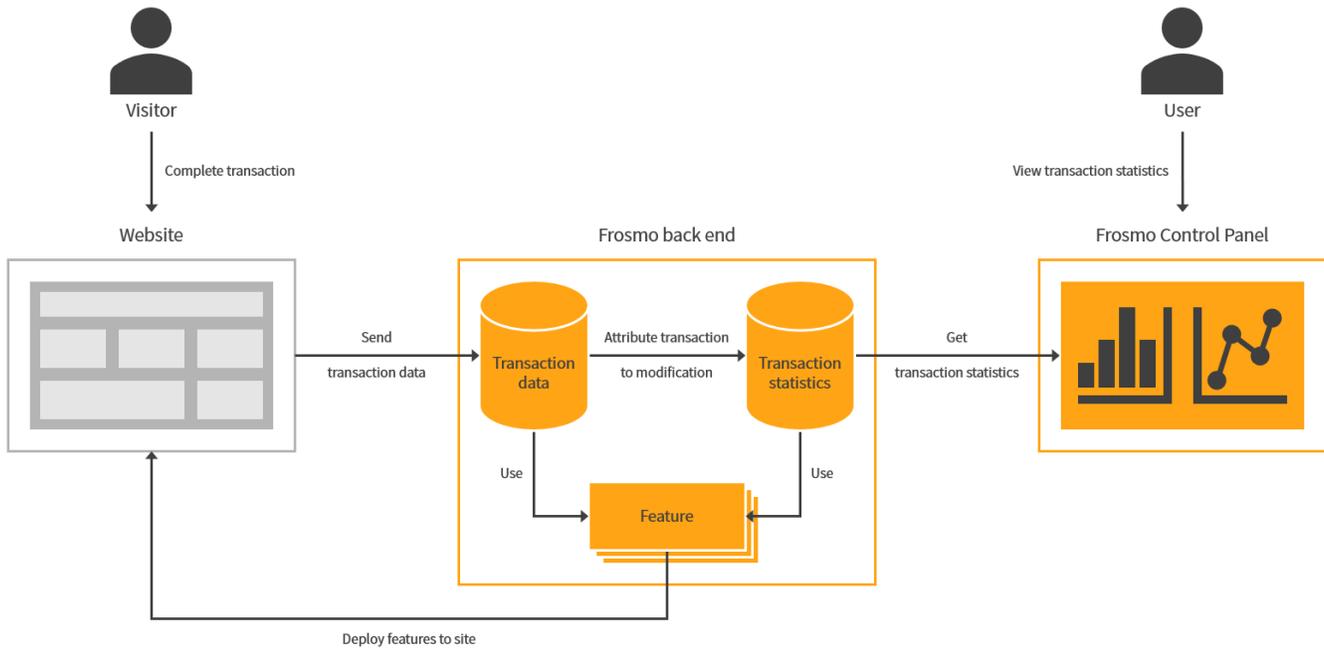


Figure: Transaction tracking in the Frosmo Platform (click to enlarge)

Transaction tracking generates transaction and modification statistics, which you can [view in the Frosmo Control Panel](#).

For more information about conversions and conversion attribution, see the [conversions user guide](#).



If you want to track a product conversions that do not involve a purchase, use [conversion tracking](#).

Tracking transactions with the data layer

Tracking transactions with the [data layer](#) means triggering a **transaction event** whenever a visitor successfully completes an action that qualifies as a transaction. The data you pass in the transaction event defines the transaction.



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

You can trigger transaction 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)

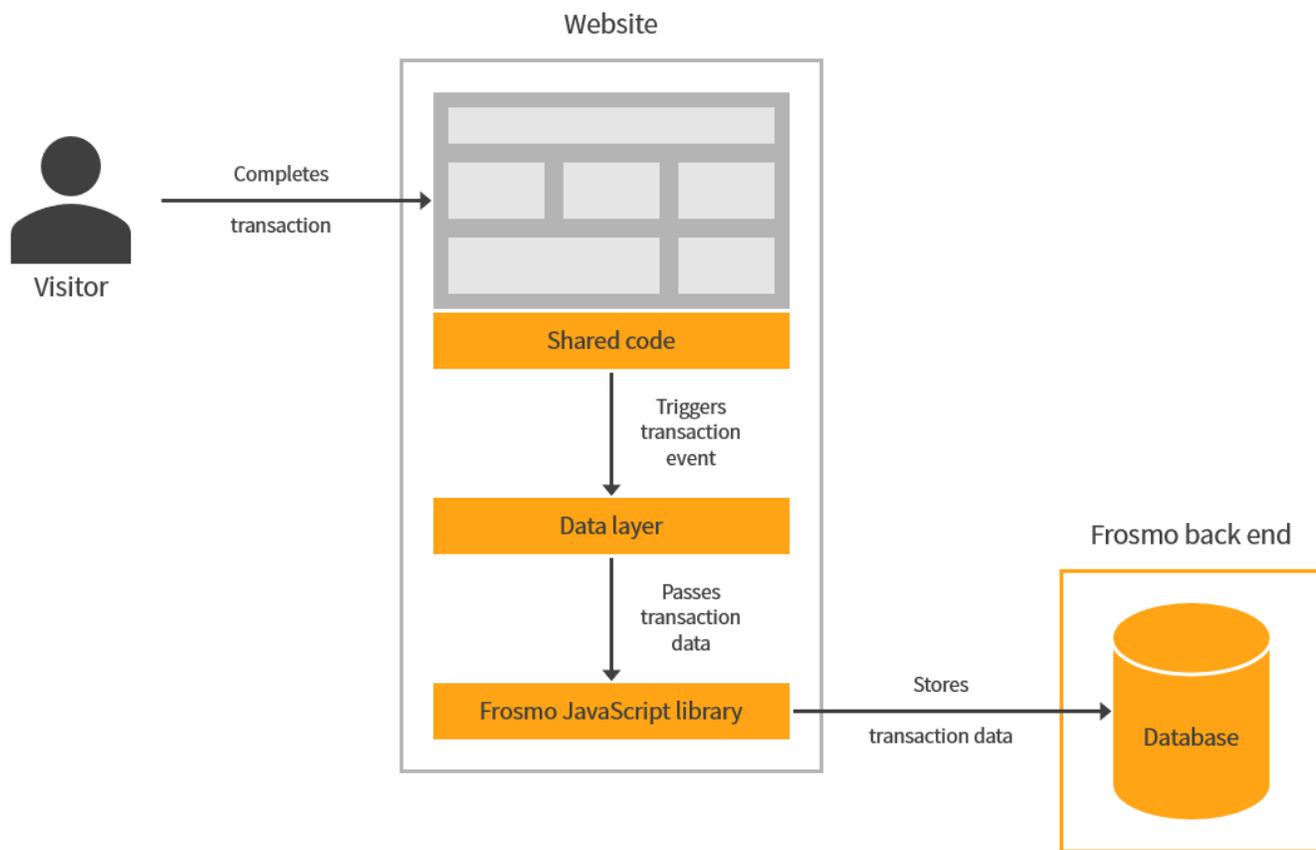


Figure: Tracking transactions by triggering a transaction event from shared code (click to enlarge)

Triggering transaction events

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

```
dataLayer.push({
  transactionProducts: [{
    id: 'string',
    name: 'string',
    price: 0,
    sku: 'string',
    /* Optional */
    quantity: 0
  }],
  /* Optional */
  transactionId: 'string',
  transactionTotal: 0
});
```

Transaction object

The transaction object contains the data of a transaction event.

Table: Transaction object properties

Property	Description	Type	Role
----------	-------------	------	------

transaction Id	<p>Unique transaction ID.</p> <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;">  The platform registers a transaction with a given ID only once. If you trigger a transaction event with the same ID multiple times, the platform ignores all but the first event. </div>	String (32)	Optional
transaction Products	<p>List of products purchased in the transaction.</p> <p>For more information about the product data, see the following table.</p>	Array of transaction product objects	Required
transaction Total	<p>Total purchase value of the transaction.</p>	Number	Optional

Table: Transaction product object properties

Property	Description	Type	Role
id	<p>Product ID.</p> <p>The product ID is the same value you pass in the <code>frosmoProductId</code> property of a product object when tracking products.</p> <p>This property is optional if <code>name</code> or <code>sku</code> is defined. You must define at least one of the three.</p> <div style="border: 1px solid #ffc107; padding: 5px; margin-top: 10px;">  If you do not use the proper product ID as the property value, the platform has no way of mapping the purchased product to the corresponding product in the Frosmo product database. For example, if you wanted to create a product recommendation based on the transactions, the recommendation engine would not be able to map the purchased products to tracked products and would therefore fail to return the product data for the transactions. </div>	String (128)	Required /Optional
name	<p>Product name.</p> <p>The product name is the same value you pass in the <code>frosmoProductName</code> property of a product object when tracking products.</p> <p>This property is optional if <code>id</code> or <code>sku</code> is defined. You must define at least one of the three.</p>	String (128)	Required /Optional
price	<p>Unit price used for the product in the transaction.</p>	Number	Required
quantity	<p>Number of items purchased in the transaction.</p>	Number	Optional
sku	<p>Stock keeping unit (SKU) code for the product.</p> <p>This property is optional if <code>id</code> or <code>name</code> is defined. You must define at least one of the three.</p>	String (128)	Required /Optional

Transaction examples

Example: Triggering a transaction event with one product

```
dataLayer.push({
  transactionId: '1010101010',
  transactionProducts: [{
    id: '123456',
    name: 'Cheetah Plushy',
    price: 9.99,
    quantity: 3
  }]
});
```

Example: Triggering a transaction event with two products

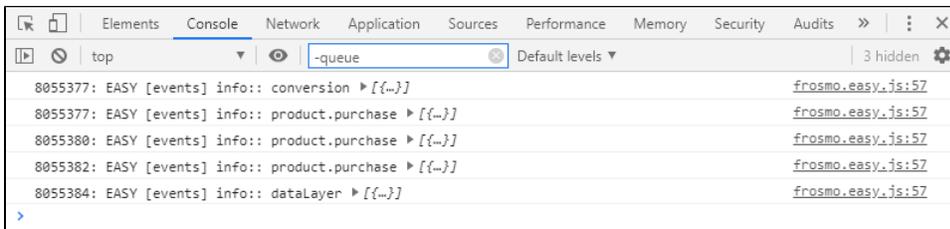
```
dataLayer.push({
  transactionId: '1010101011',
  transactionProducts: [{
    id: '123456',
    name: 'Cheetah Plushy',
    price: 9.99,
    quantity: 3
  }, {
    id: '123457',
    name: 'Lion Plushy',
    price: 19.99,
    quantity: 2
  }]
});
```

Testing transaction tracking

To test that transactions 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 transactions are tracked. If transaction events are successfully triggered with the data layer, the browser console displays the following messages for each event:
 - EASY [events] info:: conversion (contains the transaction data parsed from the transaction object)
 - EASY [events] info:: product.purchase (contains the ID and price for a single purchased product item)
 - EASY [events] info:: dataLayer (contains the transaction object passed to the data layer)

The following figure shows an example of the transaction event messages.



i Transaction events log a dedicated `product.purchase` message for each individual purchased item. For example:

- If a transaction contains three copies of one product, the transaction event logs three `product.purchase` messages.
- If a transaction contains three copies of one product and two copies of another product, the transaction event logs five `product.purchase` messages.

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

