

Tracking products and transactions

This example relies on [product data](#) and [transaction data](#) for generating recommendations. You must therefore first get that data into the Frosmo back end. In this example, you use the [Frosmo data layer](#) to track products and transactions directly on the site.



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

You can use the data layer either from your site's source code or in modification content. In this example, you implement the tracking with modifications.

To track products and transactions:

1. [Implement product tracking with the data layer](#).
2. [Implement transaction tracking with the data layer](#).

The following figure shows you an overview of the workflow for tracking products and transactions (click to enlarge).



Figure: Workflow for tracking products and transactions

Tracking products

In this example, you track products with a modification placed on every product page on the site. When a visitor navigates to a product page, the modification loads, scrapes the product data from the page DOM, and uses the data layer to pass that data to the Frosmo back end. The modification is only used for collecting the data, so it does not display any content on the page.

For the purpose of this example, you can identify a product page by the `controller=product` query parameter in the page URL.

To track products with a modification:

1. [Create a placement for the modification](#).
2. [Create the modification](#).
3. [Test the data tracking](#).

Creating the product tracking placement

To create the placement:

1. In the Control Panel, select **Modifications > Placements**.
2. Click **Create placement**.
3. Define the following settings:
 - **Name:** Enter a descriptive name, such as "Tracking - Product Page".
 - **Target element:** Select **CSS selector**, and enter "body" as the selector name.
 - **Display method:** Select **Before content**.
 - **URL matcher:** Click **Add new rule**. Since only the product page URL contains the query parameter `controller=product`, define the following rule: "Page query selector controller is exactly product". This means that the placement is valid for the page only when the page URL contains the query key "controller" with the value "product".

For more information about the URL matcher, see [Creating and editing a placement](#).

The target element, display method, and URL matcher together place the modification as the first child element of the `body` element on a product page. Since the modification does not add any visible content to the page, the location of the modification doesn't really matter.

× Create placement

Name
Tracking - Product Page

Description - *optional*

Target element [?]
CSS selector

Display method [?]
Before content

URL matcher [?] - *optional*

Page query selector
is exactly

4. Click **Save**.

You have created the placement.

For more information about placement settings, see [Creating and editing a placement](#).

Creating the product tracking modification

To create the modification:

1. In the Control Panel, select **Modifications > Overview**.
2. Click **Create modification**.
3. Enter a descriptive name for the modification, such as "Track Products", select **Basic** as the modification type, and deselect **Enable comparison group**.

Modification name

The name is not displayed to the site visitors

Modification type

Basic
Cached
Automatic
Recommendation

Enable comparison group

The comparison group is a useful feature for production modifications, so do not disable it by default. Here, the comparison group is disabled to keep the example as simple as possible. For more information about the comparison group, see [Default configuration and data tracking for a visitor](#).

4. Click **Create modification**.

5. Create the modification content:

a. Select **Content**.

b. Expand variation 1.

c. In the content field, enter the code that implements the product tracking. Use the following example code as a basis.

The example code first checks whether the data layer is available on the page. If the data layer is available, the code scrapes the product data from the page and then sends the data to the Frosmo back end by calling `dataLayer.push()` with a [product object](#).

```

<script>
  /**
   * If the data layer is not available, throw an error, which stops the script from executing.
   * The error will show up in the error tracking UI of the Frosmo Control Panel.
   */
  if (!Array.isArray(window.dataLayer)) {
    throw new Error('Missing or invalid dataLayer object. Expected an array instead of "' +
typeof window.dataLayer + '".');
  }

  /**
   * The data layer is available, so save the product data.
   */
  saveProductData();

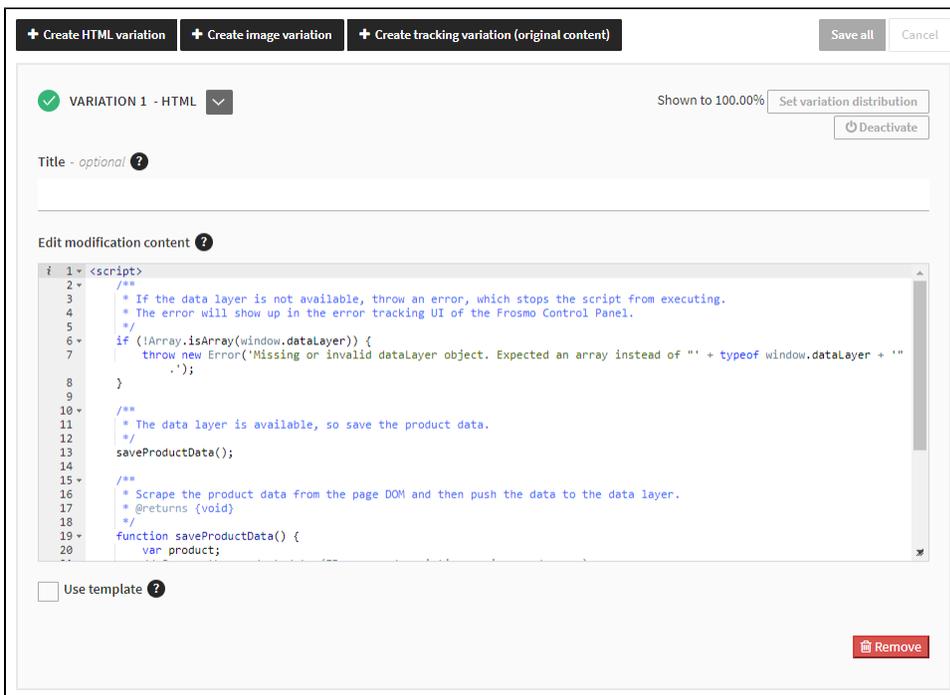
  /**
   * Scrape the product data from the page DOM and then push the data to the data layer.
   * @returns {void}
   */
  function saveProductData() {
    var product;
    // Scrape the product data (ID, name, description, price, and so on).
    // ...

    // Use the data layer to pass the product data to the Frosmo back end.
    // Note! The following code assumes that the product data has been scraped into
    // the "product" object variable.
    window.dataLayer.push({
      frosmoProductId: product.id,
      /* Optional */
      frosmoProductCategory: product.category_name,
      frosmoProductDescription: product.description,
      frosmoProductDiscountPrice: product.unit_sale_price,
      frosmoProductImage: product.image_url,
      frosmoProductName: product.name,
      frosmoProductPrice: product.unit_price,
      frosmoProductUrl: product.url
    });
    console.log('Product data saved.');
```

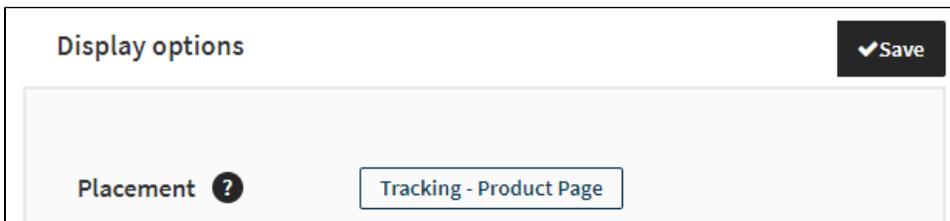


The code omits the details of scraping the product data, since these depend entirely on how the product page is built. The code also assumes that a specific set of product data fields is available on the product page. The set of data available on your site's product pages may be different.

- d. Click **Activate** for variation 1.
- e. Click **Save all**.



6. Define the display options for the modification:
 - a. Select **Display Options**.
 - b. Click **Select placement**.
 - c. Select the placement you created above, and click **Select**.
 - d. Click **Save**.



7. Disable basic conversion tracking for the modification:
 - a. Select **Analytics**.
 - b. At the bottom of the page, deselect **Enable display and click conversion tracking for this modification**.

Since the modification is used purely for collecting data, and since the modification displays no content that visitors can view or click, you do not want to attribute conversions to the modification based on the clicks, displays, and true displays it receives. For more information about conversions and modifications, see [Introduction to conversion tracking](#).

8. Click **Activate** for the modification.

You have created the modification. Products are now tracked on your site.

For more information about modification settings, see [Creating and editing a modification](#).

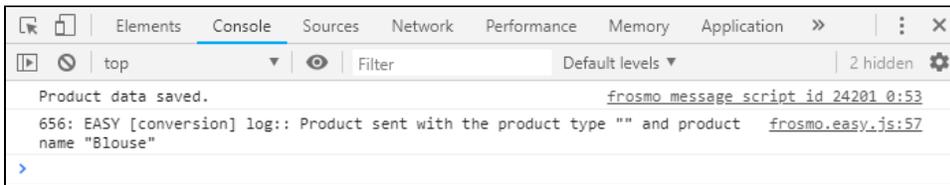
Testing the product tracking

To test that products are correctly tracked:

1. Go to the site.
2. Open the browser console.
3. Enable Frosmo Core console logging by making the following function call:

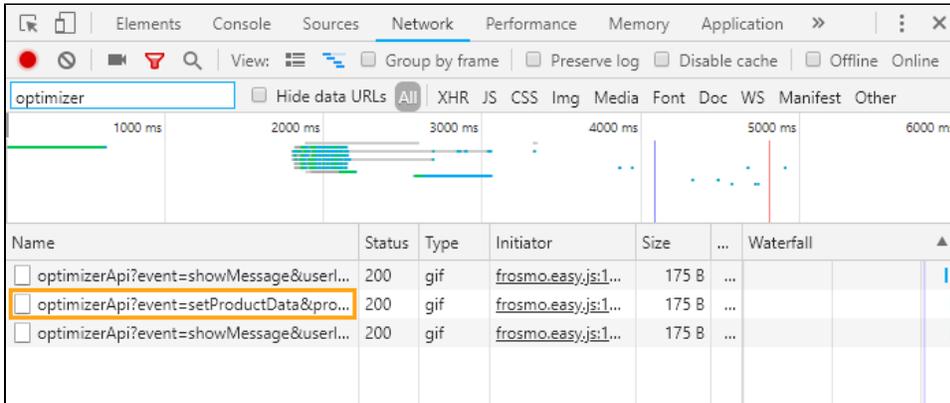
```
frosmo.easy.console.setEnabled(true);
```

4. Go to a product page. If the product data is successfully collected, the console displays the following messages:



i In the above screenshot, the first message comes from the `console.log()` call in the modification content, while the second messages comes from Frosmo Core. Even if you omit manual console logging in the modification, you still get the second message.

- If you want more details on the data layer push, 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.



✓ To show only Optimizer API requests, filter the requests list by "optimizer".

Tracking transactions

In this example, you track transactions (product purchases) with a modification placed on the order confirmation page, which is assumed to contain the necessary transaction data. When a visitor completes a purchase, they are directed to the order confirmation page, the modification loads, scrapes the transaction data from the page DOM, and uses the data layer to pass that data to the Frosmo back end. The modification is only used for collecting the data, so it does not display any content on the page.

For the purpose of this example, you can identify the order confirmation page by the `controller=confirmation` query parameter in the page URL.

To track transactions with a modification:

1. [Create a placement for the modification.](#)
2. [Create the modification.](#)
3. [Test the data tracking.](#)

Creating the transaction tracking placement

To create the placement:

1. In the Control Panel, select **Modifications > Placements**.
2. Click **Create placement**.
3. Define the following settings:
 - **Name:** Enter a descriptive name, such as "Tracking - Order Confirmation Page".
 - **Target element:** Select **CSS selector**, and enter "body" as the selector name.
 - **Display method:** Select **Before content**.
 - **URL matcher:** Click **Add new rule**. Since only the order confirmation page URL contains the query parameter `controller=confirmation`, define the following rule: "Page query selector controller is exactly confirmation". This means that the placement is valid for the page only when the page URL contains the query key "controller" with the value "confirmation".

For more information about the URL matcher, see [Creating and editing a placement](#).

The target element, display method, and URL matcher together place the modification as the first child element of the `body` element on the order confirmation page. Since the modification does not add any visible content to the page, the location of the modification doesn't really matter.

✕ Create placement

Name
Tracking - Order Confirmation Page

Description - optional

Target element [?]
CSS selector

Display method [?]

URL matcher [?] - optional

Page

4. Click **Save**.

You have created the placement.

For more information about placement settings, see [Creating and editing a placement](#).

Creating the transaction tracking modification

To create the modification:

1. In the Control Panel, select **Modifications > Overview**.
2. Click **Create modification**.
3. Enter a descriptive name for the modification, such as "Track Purchases", select **Basic** as the modification type, and deselect **Enable comparison group**.

Modification name

The name is not displayed to the site visitors

Modification type

Basic
Cached
Automatic
Recommendation

Enable comparison group

The comparison group is a useful feature for production modifications, so do not disable it by default. Here, the comparison group is disabled to keep the example as simple as possible. For more information about the comparison group, see [Default configuration and data tracking for a visitor](#).

4. Click **Create modification**.
5. Create the modification content:
 - a. Select **Content**.
 - b. Expand variation 1.
 - c. In the content field, enter the code that implements the transaction tracking. Use the following example code as a basis.

The example code first checks whether the data layer is available on the page. If the data layer is available, the code scrapes the transaction data from the page and then sends the data to the Frosmo back end by calling `dataLayer.push()` with a [transaction object](#).

```

<script>
  /**
   * If the data layer is available, save the transaction data.
   */
  if (window.dataLayer) {
    saveTransactionData();
  }

  /**
   * Scrape the transaction data from the page DOM and then push the data to the data layer.
   * @returns {void}
   */
  function saveTransactionData() {
    // Scrape the transaction data.
    // For example, let's assume the data is available in a global "window.transaction"
    // object. The "window.transaction.id" property contains the transaction ID, while
    // the "window.transaction.products" property contains the product data as an array
    // of objects, with each object describing a single purchased product.

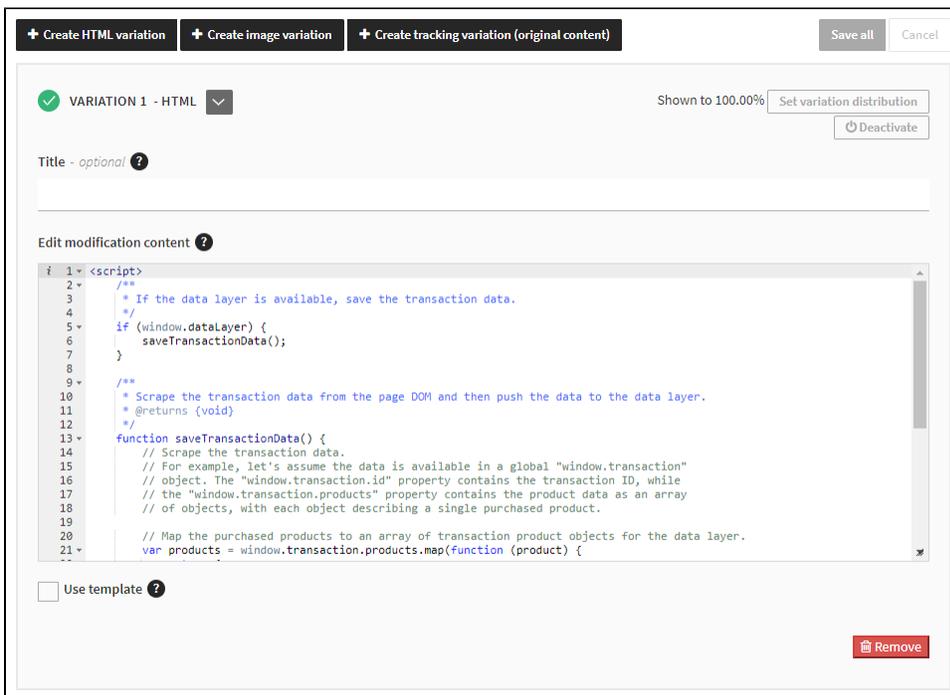
    // Map the purchased products to an array of transaction product objects for the data
layer.
    var products = window.transaction.products.map(function (product) {
      return {
        id: product.id,
        name: product.name,
        price: product.price,
        quantity: product.quantity
      };
    });

    // Use the data layer to pass the transaction data to the Frosmo back end.
    window.dataLayer.push({
      transactionId: window.transaction.id,
      transactionProducts: products
    });
    console.log('Transaction data saved.');
```



The code assumes that the order confirmation page provides a global object that conveniently contains the transaction data required for the data layer call. Getting that same data on your site may require a completely different approach, especially if the data is not available on the order confirmation page. For example, depending on how your site is implemented, you may need to store the data at an earlier point in the purchase funnel, and then retrieve the data for the data layer push on the order confirmation page.

- d. Click **Activate** for variation 1.
- e. Click **Save all**.



6. Define the display options for the modification:
 - a. Select **Display options**.
 - b. Click **Select placement**.
 - c. Select the placement you created above, and click **Select**.
 - d. Click **Save**.



7. Disable basic conversion tracking for the modification:
 - a. Select **Analytics**.
 - b. At the bottom of the page, deselect **Enable display and click conversion tracking for this modification**.

Since the modification is used purely for collecting data, and since the modification displays no content that visitors can view or click, you do not want to attribute conversions to the modification based on the clicks, displays, and true displays it receives. For more information about conversions and modifications, see [Introduction to conversion tracking](#).

8. Click **Activate** for the modification.

You have created the modification. Transactions are now tracked on your site.

For more information about modification settings, see [Creating and editing a modification](#).

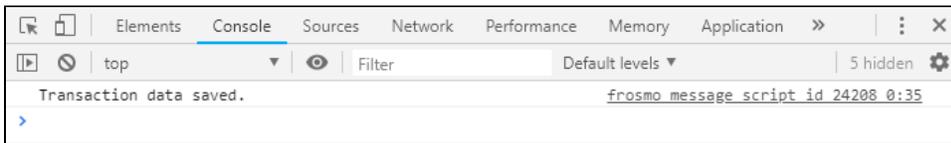
Testing the transaction tracking

To test that transactions are correctly tracked:

1. Go to the site.
2. Open the browser console.
3. Enable Frosmo Core console logging by making the following function call:

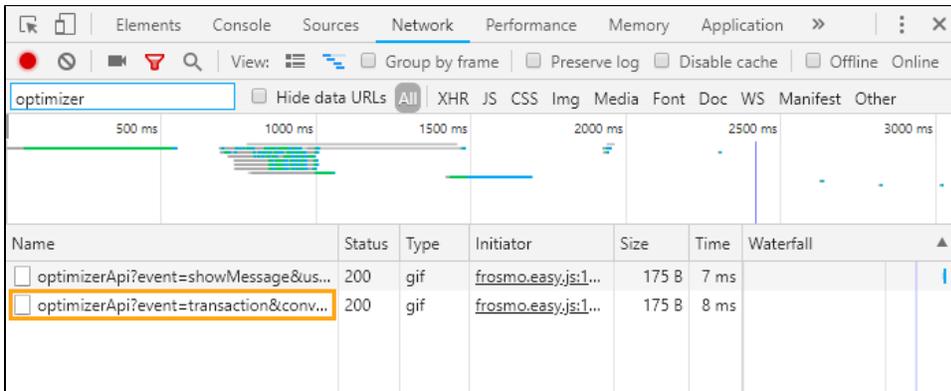
```
frosmo.easy.console.setEnabled(true);
```

4. Go to the order confirmation page by completing a purchase or by other means. If the transaction data is successfully collected, the console displays the following message:



i In the above screenshot, the message comes from the `console.log()` call in the modification content. If you omit manual console logging in the modification, you need to check the request details in the **Network** tab (see the next step).

5. If you want more details on the data layer push, 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.



✓ To show only Optimizer API requests, filter the requests list by "optimizer".