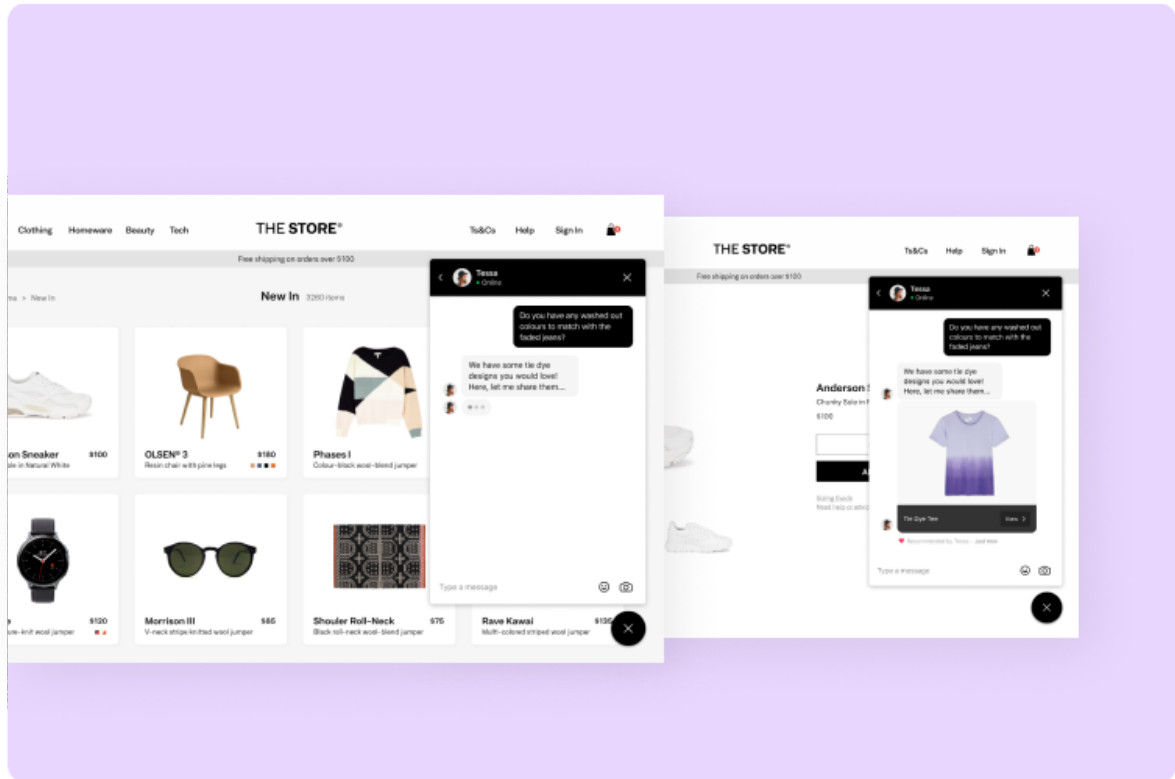


Developer Docs

Getting Started

Welcome to the Developer Docs for Virtual Shopping. We've put together this simple guide to help you get set-up. When you're done, customers will be able to virtually shop with your store team.



The Virtual Shopping On-site chat feature

What you need to know

Virtual Shopping embeds easily on any website with a lightweight JavaScript code snippet. It will only appear after important website content because it loads asynchronously, and it won't interfere with any other on-page content, code, or SEO because it's in an iframe. It's also extendible so you can listen to and interact with it using the Virtual Shopping API for more advanced setups.

Now that we've covered the basics, you're ready to start setting up Virtual Shopping in 5 key steps.

⚠ Before you begin

Make sure to consider whether other testing or development work streams are taking place on your website to avoid delays to launch.

Tests and changes that occur at the same time as the Virtual Shopping implementation risk overwriting changes and interfering with operations. If there are any other activities taking place in the same window, please let us know so we can plan around it.

Step 1. Implement Virtual Shopping

Add Virtual Shopping correctly to a website via Javascript, including setting up events tracking and

...and Virtual Shopping correctly to a remote marketplace, including setting up event tracking and meeting requirements for cookies.



Implement Virtual Shopping



Timings Guide!

Suggested Brand Resource

1 Web Developer (with access to your staging site)

Suggested Brand Timeline

2 days

Step 2. Configure On-Site Chat

Set up On-site chat to provide shoppers with the optimum experience.



Configure Virtual Shopping



Timings Guide!

Suggested Brand Resource

1 Product Feed Administrator

Suggested Brand Timeline

0.5 days

Step 3. Extend Virtual Shopping

Go beyond the core On-site Chat implementation to do more.




Extend Virtual Shopping

Step 4. Test Virtual Shopping

Once you're all set up, make sure Virtual Shopping is working correctly.



Test Virtual Shopping

 Timings Guide!

Suggested Brand Resource

1 Web Developer

1 Quality Assurance Engineer

Suggested Brand Timeline

1.5 days

Step 5. Ready Stores

Ensure your store teams are technically set up for Virtual Shopping.



Ready Stores

Implement Virtual Shopping

Overview

Implement Virtual Shopping correctly

In this section, you'll find out how to implement Virtual Shopping correctly on your website. Read this section to learn:

How to add the Javascript code snippet in all the right places.

How to track Shopper Events using the Virtual Shopping Client API.

The wide range of Shopper Events, why they're good and what they do.

Cookie requirements are met for displaying Virtual Shopping aligning with shoppers' data preferences.



Add the Virtual Shopping tag



Track Shopper Events



Cookie requirements

Add the Virtual Shopping tag

How do you add Virtual Shopping to your website?

For Virtual Shopping to work on your website, you will need to add a small piece of JavaScript to your website pages. Adding the code snippet to every page of your website helps ensure:

- ✔ **On-site Chat displays correctly** in the right places
- ✔ **Experts can continue chats** as shoppers navigate to other pages
- ✔ **Every sale supported by Virtual Shopping is tracked** from chat to checkout

Note that you can choose to deploy the JavaScript via Google Tag Manager or Tealium.

✔ We recommend you add the JavaScript code snippet to every page of your website. This is so that chats can follow shoppers from page to page after an initial chat is started and so that sales are tracked correctly.

You do not have to add the Virtual Shopping tag to sensitive pages in your checkout journey but the tag does need to be deployed to the transaction confirmation / complete page to track sales.

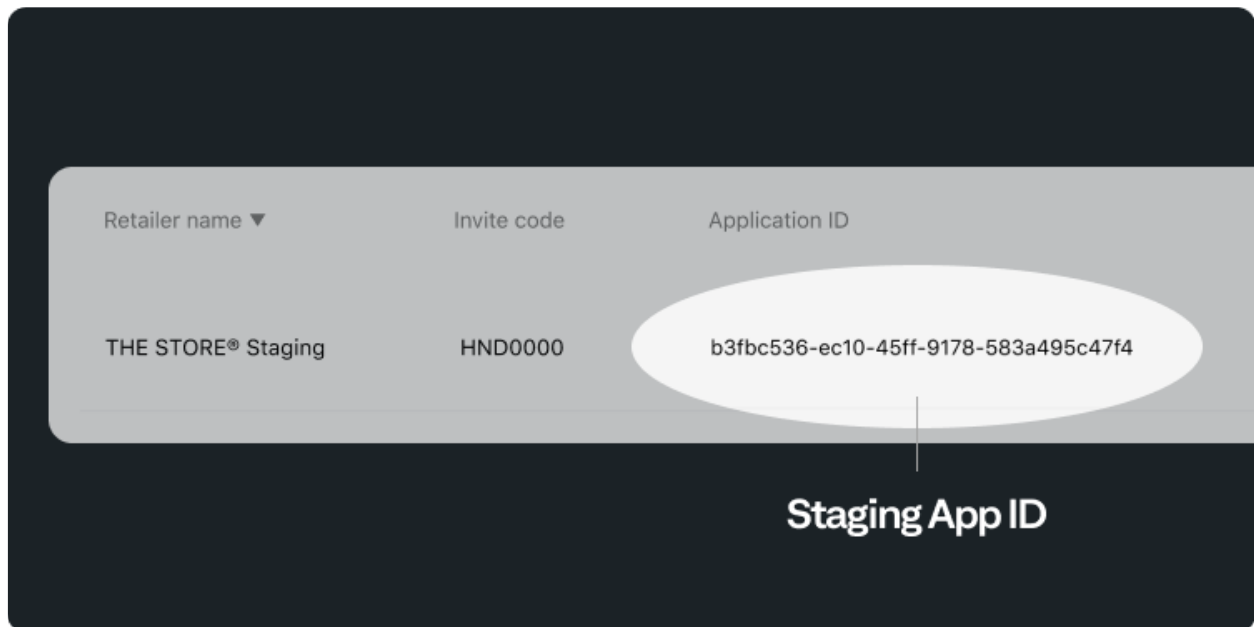
Later in setup, you can [specify where On-site chat will appear](#) on your website with URL whitelisting and blacklisting. The best places to show the Virtual Shopping icon are on your website's product and category pages.

JavaScript code snippets for Staging and Production

Ahead of implementation, we will send you two App IDs for each website region. In each case, one App ID is for your Staging environment and one for your Production environment.

1. **Staging App ID** - add this to your Staging website pre-production or any other environment for testing which is not public and best reflects the live environment.
2. **Production App ID** - add this to your live production environment. Klarna will need to complete your production website's configuration and confirm that everything is set up correctly before the JavaScript is deployed. This step will usually take place at the end of the integration process just before going live.

IDs look like this:



i **Why 2 IDs?** The 2 IDs relate to different instances on Virtual Shopping's backend. Having different instances of Virtual Shopping means you can test on your staging website and leave your production website clear for live data.

How to add the JavaScript code snippets

On both staging and production websites, copy the corresponding JavaScript code snippet and paste them just before the closing body tag on every page of your website replacing KLARNA-XXXX with the unique Application ID:

```
<script>>window.HeroWebPluginSettings = { applicationId: "KLARNA-XXXX" };</script>  
<script>(function(i,a,m,h,e,r,o){i.HeroObject=e;i[e]=i[e]||function(){(i[e].q=i[e].q|| []).pu
```

! Although Virtual Shopping is often focused on PDP and category-level pages, you should deploy the tag to almost your whole site, including the post-purchase / checkout complete page so that sales influenced by Virtual Shopping can be accurately tracked.

Virtual Shopping does not need to be deployed to the more sensitive pages in your checkout journey.

Testing the Virtual Shopping tag

Once the Virtual Shopping team has completed setting up your configurations on their side, it's good practice to test that your Virtual Shopping Tag implementation is set up correctly so you can feel confident ahead of launch. When you're ready to do this, follow this guide:



Test the Virtual Shopping tag

Track Shopper Events

Set up Virtual Shopping to track shopper behaviour

Virtual Shopping can empower store team members with context on how shoppers are behaving on your website to support their conversations. For this to work, you will need to set up Virtual Shopping to track the specific actions shoppers take on your website.

In this section we'll learn about:

- ✔ **What the Virtual Shopping API** is and how it works.
- ✔ **What Shopper Events are**, and the different types.
- ✔ **What eCommerce subfields are** and the different types.

What is the Virtual Shopping Client API?

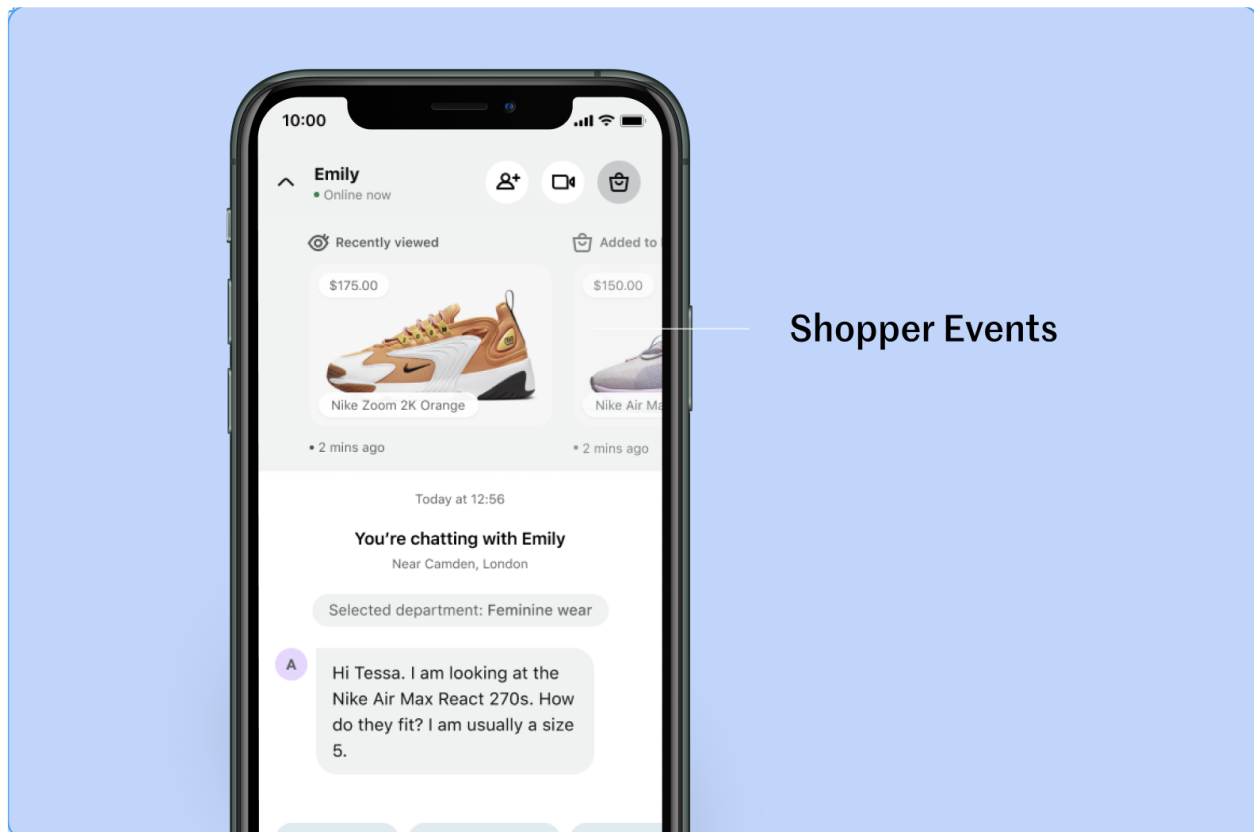
Once you have added the Virtual Shopping JavaScript code snippet to your pages, you will then have access to the Virtual Shopping API. The Virtual Shopping Client API interacts with and listens to Shopper Events and sends tracking events to you. Shopper Events can then be easily set up on your page template.



The Virtual Shopping Client API should only be accessed after the On-site Chat feature has been added to your website. Learn how to do this [here](#).

The Virtual Shopping Client API exposes a `track` method which can be used to track actions, page views, and eCommerce events across your website. Tracked Shopper Events are then processed and shown to store teams to inform them of the customer's needs and their journey prior to and during the chat.

What are Shopper Events?



Shopper Events

Shopper Events provide insights into the behaviour of every shopper browsing your website. There are several types of Shopper Events you can track to provide associates with information:



Product View



Purchase



Basket Add



Basket Remove



Category View



Search



Make sure 'Products view' and 'Purchased' Shopper Event tracking is set up at launch.

Virtual Shopping will not work properly without these events in place. Tracking for other Shopper Events can always be set up later on.

eCommerce Subfields

Each Shopper Event needs to include eCommerce subfields so that Virtual Shopping can track all the key information to show in each event. Review the full list of required subfields here:



eCommerce Subfields

Testing Shopper Events

Once the Klarna team has completed setting up your configurations on their side, it's good practice to test that your Shopper Event tracking is set up correctly so you can feel confident ahead of launch. Find out how to do this here:



Test Shopper Event tracking

Product View

Shopper Event

Viewed item



The Product View Shopper Event gives store experts who are about to chat to shoppers, context on which products they have been viewing.

The Product View Shopper Event should be set up to fire on **all PDP pages**.



Product View Shopper Event tracking must be set up at launch because it is fundamental to using Virtual Shopping.

Product View fields:

Name	Value	Type	Required	Description
type	ecommerce:detail	text	yes	The type of tracking
products	See eCommerce products subfields	array	yes	The product details



Key things to check

The Product ID is a required field and must reflect what was sent to the Store app in the product feed.

- The Product ID reflects the individual product - not the group ID.
- The price is written as a number not a String.
- The entire product array cannot be enclosed within a String.

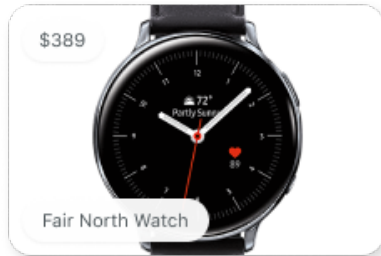
Product View example:

```
hero("track", {
  type: "ecommerce:detail",
  products: [
    {
      id: "314-7216-102",
      name: "Tennis Shoe Classic - Size 10",
      image: "https://example.com/images/314-7216-102.jpg",
      price: 70.99,
      currency: "USD",
      brand: "Plausible Co.",
      category: "Footwear > Sports > Tennis",
      location: "https://example.com/products/314-7216"
    }
  ]
});
```

Purchase

Shopper Event

☰ Purchased item



The Purchase Shopper Event gives store experts who are about to chat to shoppers, context on which products they have recently purchased.

The Purchase Shopper Event should be set up to fire on **all purchase complete/ thank you pages**



Purchase Shopper Event tracking must be set up at launch because it is fundamental to using Virtual Shopping.

Purchase fields:

Name	Value	Type	Required	Description
type	ecommerce:purchase	text	yes	The type of tracking
purchase	See eCommerce purchase subfields	object	yes	The purchase details
products	See eCommerce products subfields	array	yes	The product details

✓ Key things to check

Purchase event IDs are correct

- Every purchase event has a unique ID that can be used to record the transaction.
- The product ID `products: [` reflects what was sent to Klarna in the product feed.
- Product ID reflects the individual product - not the group ID.
- Subtotal, total, tax, shipping cost, quantity & price is written as a number - not a String.

Shipping costs are excluded

- Any shipping costs are excluded from the subtotal sent to Klarna as this is the figure that any associate commission will be based on.
- In the example below, the total field is `74.98` which is the total price the shopper has been charged including tax and shipping costs. The subtotal of `70.99`, which excludes the shipping cost of `3.99` but includes tax, is the total that will be recorded against that associate for the sale.

Tax preference is decided

- It is up to you whether or not to include tax within the subtotal sent to Klarna. Please note associate commission will be based on the sub-total value.

Purchase example:

```
hero("track", {
  type: "ecommerce:purchase",
  purchase: {
    id: "abcd",
    subtotal: 70.99,
    currency: "USD",
    total: 74.98,
    tax: 14.20,
    shippingCost: 3.99,
    shippingMethod: "Store pickup",
    coupon: "SUMMER_SALE"
  },
  products: [
    {
      id: "314-7216-102",
      name: "Tennis Shoe Classic - Size 10",
      image: "https://example.com/images/314-7216-102.jpg",
      price: 70.99,
      currency: "USD",
      quantity: 1,
      brand: "Plausible Co.",
      category: "Footwear > Sports > Tennis",
      location: "https://example.com/products/314-7216"
    }
  ]
});
```



Basket Add

Shopper Event

 Added to basket



The Basket add Shopper Event gives store experts who are about to chat to shoppers context on which products are in the shopper's basket.

 **Basket add Shopper Event tracking is not recommended for the initial launch.** This shopper event is a 'nice to have' addition and can be set up after launch. 'Product View' and 'Purchase' Shopper Event tracking must be set up at launch because these events are fundamental to using Virtual Shopping.

Basket add fields:

Name	Value	Type	Required	Description
type	<code>ecommerce:add</code>	text	yes	The type of tracking
products	See eCommerce products subfields	array	yes	The product details

Key things to check

Product IDs are correct

- There is a Product ID for every entry.
- The Product ID reflects what was sent to Klarna in the product feed.
- The Product ID reflects the individual product - not the group ID.
- The quantity, basket quantity, and price are written as a number, not a String.
- Please note: the entire product array cannot be enclosed within a String.

Basket add example:

```
hero("track", {  
  
  type: "ecommerce:add",  
  products: [  
    {  
      id: "314-7216-102",  
      quantity: 1,  
      basketQuantity: 1,  
      name: "Tennis Shoe Classic - Size 10",  
      image: "https://example.com/images/314-7216-102.jpg",  
      price: 70.99,  
      currency: "USD",  
      brand: "Plausible Co.",  
      category: "Footwear > Sports > Tennis",  
      location: "https://example.com/products/314-7216"  
    }  
  ]  
});
```


Basket Remove

Shopper Event

 Removed from basket



The basket remove Shopper Event gives store experts who are about to chat to shoppers, context on which products have been removed from a shopper's basket.

 **Basket remove Shopper Event tracking is not recommended for initial launch.** This shopper event is a 'nice to have' addition and can be set up after launch. 'Product View' and 'Purchase' Shopper Event tracking must be set up at launch because these events are fundamental to using Virtual Shopping.

Basket remove fields:

Name	Value	Type	Required	Description
type	<code>ecommerce:remove</code>	text	yes	The type of tracking
products	See eCommerce products subfields	array	yes	The product details

Key things to check

Product IDs are correct

- Product IDs are required fields and reflect that which was sent to Klarna in the product feed.
- The Product ID reflects the individual product - not the group ID.
- The quantity, basket quantity, and price are written as a number - not a String.
- Please note: the entire product array cannot be enclosed within a String.

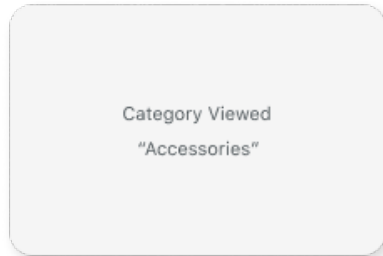
Basket remove example:

```
hero("track", {
  type: "ecommerce:remove",
  products: [
    {
      id: "314-7216-102",
      quantity: 1,
      basketQuantity: 0,
      name: "Tennis Shoe Classic - Size 10",
      image: "https://example.com/images/314-7216-102.jpg",
      price: 70.99,
      currency: "USD",
      brand: "Plausible Co.",
      category: "Footwear > Sports > Tennis",
      location: "https://example.com/products/314-7216"
    }
  ]
});
```

Category View

Shopper Event

Viewed



The Category view Shopper Event gives store experts who are about to chat to shoppers, context on which product categories they have been viewing on your website.

! **Category view Shopper Event tracking is not recommended for initial launch.** This shopper event is a 'nice to have' addition and can be set up after launch. 'Product View' and 'Purchase' Shopper Event tracking must be set up at launch because these events are fundamental to using Virtual Shopping.

Category view fields:

Name	Value	Type	Required	Description
type	event	text	yes	The type of tracking
action	category-view	text	yes	The type of interaction
value	-	text	yes	The name of the category
location	-	text	yes	URL of the page being tracked

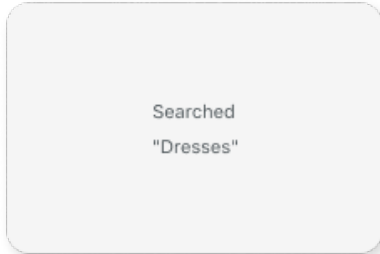
Category view event example:

```
hero("track", {  
  type: "event",  
  action: "category-view",  
  value: "Accessories",  
  location: "https://example.com/categories/accessories"  
});
```

Search

Shopper Event

🔍 Searched



The Search Shopper Event gives store experts who are about to chat with shoppers, context on which products shoppers have recently searched for on your website.

! **Search Shopper Event tracking is not recommended for the initial launch.** This shopper event is a 'nice to have' addition and can be set up after launch.

'Product View' and 'Purchase' Shopper Event tracking must be set up at launch because these events are fundamental to using Virtual Shopping.

Search event fields:

Name	Value	Type	Required	Description
type	event	text	yes	The type of tracking
action	search	text	yes	The type of interaction
value	-	text	yes	The value being searched
location	-	text	yes	URL of the page being tracked

Search event example:

```
hero("track", {  
  type: "event",  
  action: "search",  
  value: "bedding sets",  
  location: "https://example.com"  
});
```

eCommerce Subfields

eCommerce subfields are the mandatory items you will need to provide Virtual Shopping with to ensure Shopper Events track all the key information.

⚠ Please note if all recommended fields have not been included during set up, Virtual Shopping will fall back on the data provided in your Product Feed.

Here is a full breakdown of required and non-required fields:

Purchase array item subfields

Name	Type	Required	Description
id	text	yes	The purchase/order
subtotal	number	yes	The total price for all products in a purchase
currency	text	yes	The purchase currency (ISO 4217)
total	number	no	The overall adjusted total calculated price for all of the products in a purchase
tax	number	no	The purchase tax
shippingCost	number	no	The purchase shipping cost
shippingMethod	text	no	The purchase shipping method
coupon	text	no	The purchase coupon

Product array item subfields

Name	Type	Required	Description
id	text	yes	The unique product or SKU (must exactly match one product in the Product Feed)
quantity	number	yes*	The product quantity
basketQuantity	number	yes**	The new basket quantity for this product
name	text	recommended	The product name
image	text	recommended	The product image URL
price	number	recommended	The product price
currency	text	recommended	The product currency (ISO 4217)
location	text	recommended	The product URL
brand	text	no	The product brand
category	text	no	Must match the category hierarchy for this product as defined in the Product Feed . E.g. Menswear > Shoes > Brogue

* Required for [Purchase](#), [Basket Add](#) and [Basket Remove](#) Shopper Events

** Required for [Basket Add](#) and [Basket Remove](#) Shopper Events

Cookie requirements

i Depending on which cookies shoppers accept on your website, it is necessary to ensure that On-site Chat is only shown to shoppers who accept the required cookies for Virtual Shopping.

How Virtual Shopping uses cookies

To provide a tailored experience, Klarna uses cookies and similar tracking technologies. [Find more details on these and how they work here.](#)

You can view full details on which cookies Klarna Virtual Shopping uses [in our privacy policy.](#)

This site uses cookies and similar technologies to function properly and to provide the services present on it, for which your consent is not required. It uses analytical cookies (own and third party) to understand and improve users' browsing experience, and profiling cookies (own and third party) to send you advertisements in line with preferences displayed while browsing online, for which your consent is required. By clicking **"Accept all cookies"**, you consent to the use of the aforementioned cookies. To manage or disable cookies, click on **"Cookie Settings"**. By clicking **"Allow only technical cookies"**, you can continue browsing and only technical cookies will be installed. For more information, please visit our **Cookie Policy**.

COOKIE SETTINGS ALLOW ONLY TECHNICAL COOKIES **ACCEPT ALL COOKIES**

Example of a cookie banner. Virtual Shopping should only be displayed if a shopper accepts the necessary cookies.

Display requirements

When a shopper browses your site, **On-site Chat must only be displayed if a shopper accepts the necessary cookies.**

If the shopper declines the use of cookies, the code for Virtual Shopping should not be initialized.

i Most of our partner merchants categorize Virtual Shopping cookies as functional cookies. However, you can decide for your site whether Virtual Shopping cookies are **strictly necessary** or **functional**.

Additionally, as is standard in cookie banners, shoppers should have the option to allow or deny cookie permissions from your cookie banner (or dedicated "manage cookie permissions" pop-up), whether asking for implied or explicit cookie consent.

Take a look at the following example of how this could be done:

×

Cookie Centre

Functionality cookies

Functionality cookies are used to activate specific functionalities of the website and a series of options (for example, language, the products selected for purchase), to improve the service provided. If you disable this type of cookie, some services or certain functions of the website might not be available or function properly, and you could be forced to change or manually enter some information or preferences each time you visit the website.

ENABLE: OFF ON

Analytical cookies

Analytical cookies, including those of third parties, allow us to understand how the website is being used by users. These cookies do not collect information regarding your identity, nor any personal

Your website may categorize cookies in different ways, and accordingly, the cookies required for Klarna Virtual Shopping may be included in a category such as "Functionality cookies". It must be possible to enable or disable these.

Accessibility requirements

Your cookie notice should meet **AA requirements** for accessibility guidelines.

As part of the launch process, we'll test your cookie notice against these guidelines and share advice on any improvements to make to better comply with accessibility standards.

Privacy policy requirements

It is also a requirement to link to [Klarna Virtual Shopping's Consumer Privacy Notice](#).

You can include this link in the following places:

- Directly in your cookie banner.
- Within your cookies permissions pop-up.
- On a dedicated cookies policy page.
- Within a dedicated privacy and cookies centre (which is accessible from your cookie banner).

Configure Virtual Shopping

Overview

Configure Virtual Shopping for the best shopper experience

In this section, you'll find out how to configure Virtual Shopping to give shoppers the optimum experience. Read on to set up:

- ✔ **Product recommendations for shoppers** by integrating your Product Feed
- ✔ **On-site chat showing at the right moments** with Whitelisting / Blacklisting and Follow Rules



Integrate Product Feed

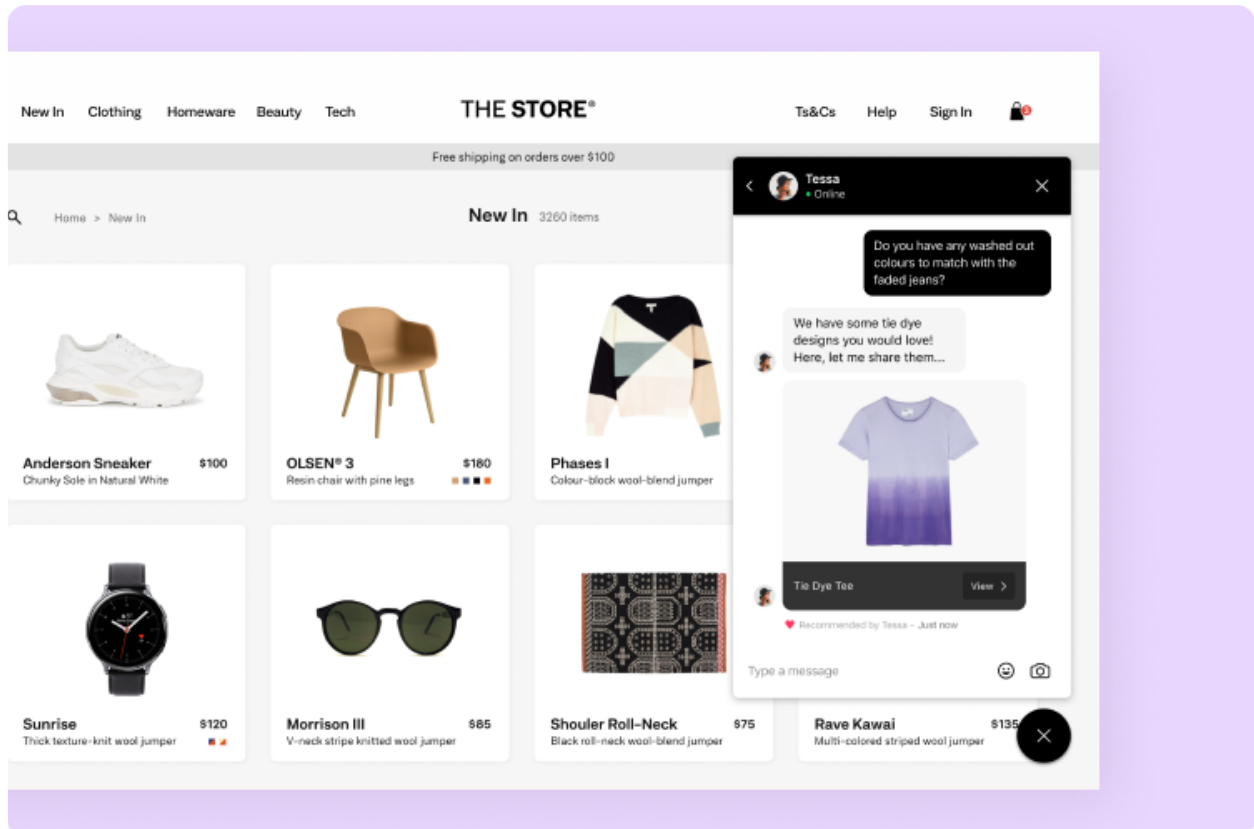


Set where Virtual Shopping appears

Integrate Product Feed

Setting up Virtual Shopping for product recommendations

With the Store App, experts can recommend products to website visitors during a chat to drive sales. These product recommendations are sent as Product Cards, which a store team member can send to a shopper during a chat.



How to integrate your Product Feed with the Store app

For this to work, the Store App needs access to a Product Feed in **CSV** or **XML** format.

To share your Product Feed with Virtual Shopping, you can share a publicly accessible URL or an SFTP for Virtual Shopping to fetch it from - or we can provide an SFTP for you to put it in.



Please ensure each product entry contains:

- Product SKU Code
- Product Title
- Product Image(s) URL
- Product URL
- Price
- Description
- Currency

- Category
- Stock Quantity



We also recommend you include other fields that contain terms your associates may use to describe or search for a product such as:

- Sale Price
- Size Variant
- Color Variant
- Additional image URLs
- Style Code or Master SKU (i.e. to group SKUs)
- Brand Name
- Barcode Number

Virtual Shopping indexes the Product Feed at least once per day so that the products available for associates to share in chats with shoppers are representative of the latest online product catalogue. Virtual Shopping can refresh the product feed more frequently if necessary.

Stock Changes

We recommend ensuring all products remain in a product feed, whether they are in stock or out of stock. Simply declare the stock quantity as 0 when out of stock, and we'll do the rest.

This ensures Virtual Shopping still has access to all products in your inventory, and experts can recommend these products.

Creating Your Product Feed

To ensure a seamless handover and quick setup, we recommend sending Virtual Shopping your Product Feed in an easy-to-handle format.



Please note to remain compliant with our cloud communications platform, no prohibited product (such as CBD) can be included within your product feed. For further information please visit our partner's [Terms of Service](#).

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	sku_code	group_sku_title	description	currency	normal_price	sale_price	stock_quantity	url	categories	image_url	gender	size	color	barcode
2	H000750006	500211 VS Sense Shampoo 280ml	Enhance your senses with VS Shampoo, the iconic product	GBP	18		17	https://example.klarna.com/products/VSSenseShampoo280	Beauty > Body > Hair > Shampoo	http://i1.laws.aws/V5/000560211-H000750006-173argel	Unisex	ONE	No Colour	5025155050910
3	H000751006	500211 VS Sense Conditioner 250ml	Enhance your colour with VS Shampoo, the iconic product	GBP	22		23	https://example.klarna.com/products/VSSenseConditioner250	Beauty > Body > Hair > Conditioner	http://i1.laws.aws/V5/000560211-H000751006-173argel	Unisex	ONE	No Colour	5025155050911
4	H000752006	500213 VS Shampoo Golden 250ml	Enhance your Golden colour with VS Shampoo, the iconic product	GBP	18		12	https://example.klarna.com/products/VSShampooGolden250	Beauty > Body > Hair > Shampoo	http://i1.laws.aws/V5/000560213-H000752006-173argel	Unisex	ONE	No Colour	5025155050912
5	H000753006	500214 VS Conditioner Golden 250ml	The ultimate VS conditioner in golden	GBP	22		6	https://example.klarna.com/products/VSConditionerGolden250	Beauty > Body > Hair > Conditioner	http://i1.laws.aws/V5/000560214-H000753006-173argel	Unisex	ONE	No Colour	5025155050913
6	H000754006	500215 VS Shampoo Red 250ml	Enhance your red locks with VS Shampoo, the iconic product	GBP	17		6	https://example.klarna.com/products/VSShampooRed250	Beauty > Body > Hair > Shampoo	http://i1.laws.aws/V5/000560215-H000754006-173argel	Unisex	ONE	No Colour	5025155050914
7	H000755006	500215 VS Conditioner Red 250ml	The ultimate VS conditioner in red	GBP	22		24	https://example.klarna.com/products/VSConditionerRed250	Beauty > Body > Hair > Conditioner	http://i1.laws.aws/V5/000560215-H000755006-173argel	Unisex	ONE	No Colour	5025155050915
8	H000756006	500217 VS Shampoo Copper 250ml	Enhance your Copper locks with VS Shampoo, the iconic product	GBP	17		11	6	https://example.klarna.com/products/VSShampooCopper250	http://i1.laws.aws/V5/000560217-H000756006-173argel	Unisex	ONE	No Colour	5025155050916
9	H000757006	500218 VS Conditioner Copper 250ml	The ultimate VS conditioner in Copper	GBP	22		22	22	https://example.klarna.com/products/VSConditionerCopper250	http://i1.laws.aws/V5/000560218-H000757006-173argel	Unisex	ONE	No Colour	5025155050917
10	H000758006	500219 VS Shampoo Tobacco 250ml	Enhance your Tobacco locks with VS Shampoo, the iconic product	GBP	18		3	3	https://example.klarna.com/products/VSShampooTobacco250	http://i1.laws.aws/V5/000560219-H000758006-173argel	Unisex	ONE	No Colour	5025155050918
11	H000759006	500220 VS Conditioner Tobacco 250ml	The ultimate VS conditioner in Tobacco	GBP	22		14	4	https://example.klarna.com/products/VSConditionerTobacco250	http://i1.laws.aws/V5/000560220-H000759006-173argel	Unisex	ONE	No Colour	5025155050919
12	H000760006	500221 VS Shampoo Chocolate 250ml	Enhance your Chocolate locks with VS Shampoo, the iconic product	GBP	18		4	4	https://example.klarna.com/products/VSShampooChocolate250	http://i1.laws.aws/V5/000560221-H000760006-173argel	Unisex	ONE	No Colour	5025155050920
13	H000761006	500222 VS Conditioner Chocolate 250ml	The ultimate VS conditioner in Chocolate	GBP	22		2	6	https://example.klarna.com/products/VSConditionerChocolate250	http://i1.laws.aws/V5/000560222-H000761006-173argel	Unisex	ONE	No Colour	5025155050921
14	H001323006	500398 VS Infusion Spray 100ml	Let VS Infuse your hair	GBP	23		19	16	https://example.klarna.com/products/VSInfusionSpray100	http://i1.laws.aws/V5/000560398-H001323006-173argel	Unisex	ONE	No Colour	5025155050922
15	H002962006	500842 VS Hair Perfume 75ml	Let VS Scent your hair	GBP	52		45	37	https://example.klarna.com/products/VSHairPerfume75ml	http://i1.laws.aws/V5/000560842-H002962006-173argel	Unisex	ONE	No Colour	5025155050923
16	H002963006	500843 VS Long Hair Perfume 75ml	Let VS Scent your long hair	GBP	52		45	75	https://example.klarna.com/products/VSLongHairPerfume75ml	http://i1.laws.aws/V5/000560843-H002963006-173argel	Unisex	ONE	No Colour	5025155050924
17	H002964006	500844 VS Water Hair Perfume 75ml	Let VS Scent your wet hair	GBP	52		33	33	https://example.klarna.com/products/VSWaterHairPerfume75ml	http://i1.laws.aws/V5/000560844-H002964006-173argel	Unisex	ONE	No Colour	5025155050925
18	H003000006	500848 VS Body Cream 200ml	The ultimate VS body Cream	GBP	130		111	6	https://example.klarna.com/products/VSBodyCream200ml	http://i1.laws.aws/V5/000560848-H003000006-173argel	Unisex	ONE	No Colour	5025155050926
19	H021314006	505529 Shirt	Live like a Klarna in this Shirt	GBP	135		4	4	https://example.klarna.com/products/Shirt	http://i1.laws.aws/V5/000565229-H021314006-173argel	Male	S	Navy	5025155050927
20	H021315006	505529 Shirt	Live like a Klarna in this Shirt	GBP	135		4	4	https://example.klarna.com/products/Shirt	http://i1.laws.aws/V5/000565229-H021315006-173argel	Male	M	Navy	5025155050928
21	H021317006	505529 Shirt	Live like a Klarna in this Shirt	GBP	135		3	3	https://example.klarna.com/products/Shirt	http://i1.laws.aws/V5/000565229-H021317006-173argel	Male	XL	Navy	5025155050929
22	H021318006	505529 Shirt	Live like a Klarna in this Shirt	GBP	135		1	1	https://example.klarna.com/products/Shirt	http://i1.laws.aws/V5/000565229-H021318006-173argel	Male	XXL	Navy	5025155050930
23	H278846006	506210 Long-Sleeve T-Shirt	Live like a Klarna in this T-Shirt	GBP	125		114	1	https://example.klarna.com/products/Long-Sleeve-T-Shirt	http://i1.laws.aws/V5/00056210-H278846006-173argel	Men		Yellow	5025155050931
24	H278847006	506210 Long-Sleeve T-Shirt	Live like a Klarna in this T-Shirt	GBP	125		4	4	https://example.klarna.com/products/Long-Sleeve-T-Shirt	http://i1.laws.aws/V5/00056210-H278847006-173argel	Men		Yellow	5025155050932
25	H282959006	506210 Long-Sleeve T-Shirt	Live like a Klarna in this T-Shirt	GBP	130		2	2	https://example.klarna.com/products/Long-Sleeve-T-Shirt	http://i1.laws.aws/V5/00056210-H282959006-173argel	Male	XL	Olive	5025155050933
26	H318347006	506210 Long-Sleeve T-Shirt	Live like a Klarna in this T-Shirt	GBP	130		3	3	https://example.klarna.com/products/Long-Sleeve-T-Shirt	http://i1.laws.aws/V5/00056210-H318347006-173argel	Male	XXL	Olive	5025155050934

Below we've assembled an example table of property names that you can use when setting up your Product Feed to send to Virtual Shopping. If using these names is not possible, the Virtual Shopping team can map your feed as part of the launch process.



For an example Product Feed containing example property fields, download the below attached example.



VS-Example-Product-feed-v1.0.csv 7KB

Text

Please scroll to the right in this table to see example values.

Property	Type	Required	Description	Example Value
<code>sku_code</code>	string	yes	The unique identifier for a specific product.	314-7216-102
<code>group_sku</code>	string	no	<p>The identifier for a group of products: each variation i.e. <code>size / colour</code> or <code>fabric</code> will have a unique identifier it's <code>sku_code</code>.</p> <p>The <code>group_sku</code> is used to ensure product variations appear under the same product in the app.</p>	314-7216
<code>title</code>	string	yes	A product's title.	Relaxed Fit T-sh
<code>description</code>	string	yes	Description of the product with all markdown or HTML removed.	Our classic 100% cotton relaxed fit shirt
<code>currency</code>	string	yes	The <code>ISO-4217</code> currency code for this product. This will generally be the same for all products in a feed.	GBP
<code>normal_price</code>	number	yes	How much the product usually costs.	99.99
<code>sale_price</code>	number	no	If the product is on sale, how much it costs while on sale.	79.99
<code>stock_quantity</code>	number	yes	How many units of this product are currently in stock. You can supply a <code>1 / 0</code> for in and out of stock, respectively.	34

<code>url</code>	URL	yes	A link to the product's product detail page.	https://example.c/products/01234,DefGH
<code>categories</code>	string	yes	The category breadcrumb trail for this product, as a delimited string.	Ladies fashion > Jeans & Trouser Skinny jeans
<code>image_url</code>	URL	yes	A link to the main image for this product.	https://example.c/products/01234,DefGH/images/01.jpg
<code>alternate_image</code>	URL	no	<p>One or more links to images for this product.</p> <p>CSV If you have multiple additional images use columns with an incremented number e.g. <code>alternate_image_1</code> <code>alternate_image_2</code></p> <p>XML Wrap each URL in the same tag e.g. <code><alternate_image>URL_1</alternate_image></code> <code><alternate_image>URL_N</alternate_image></code></p>	https://example.c/products/01234,DefGH/images/02.jpg
<code>gender</code>	string	no	If you have gender specific products, you can supply a value for gender.	unisex
<code>size</code>	string	no	Include for products that have size variations.	XX-Small
			Include for products	

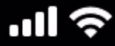
color	string	no	that have colour variations.	Navy
fit	string	no	Include for products that have fit variations.	Relaxed
fabric	string	no	Include for products that have fabric information.	98% Cotton, 2% Elastane
barcode	string	no	Include if you need to show barcode information.	0100234567890
collar	string	no	Include for products that have collar variations.	Point
feature	string	no	Include for products that have feature information.	Hidden Pocket

Representing discounted/sale prices

If a product is currently on sale, we can display it in the Store App for Product Experts (not to shoppers) with "strikethrough" pricing. If you provide a `sales_price` value for a product, we will display the `normal_price` crossed-out, and the `sale_price` in its place.

For example:

```
"normal_price": 449.00,
"sale_price": 369.00,
"currency": "GBP",
```



09:41

98%



Watch presentation box



Watch presentation box

£369.00 ~~£449.00~~

Product Details

This attractive leather and suede box provides safe storage and an attractive presentation for your horological collection.

Item ID: 021-0477

[Visit Webpage](#)

A product showing in the Store App for Products with "strikethrough" sale pricing

Product Feed error reports

We provide you with error reporting for your product feeds. This will help you ensure product feeds are regularly updated without any disruption.

Get notified of any issues with your Product Feed.

Virtual Shopping indexes the Product Feed at least once a day to ensure your product experts have access to your latest online offering. If there are any issues, which means indexing fails, your designated contacts will be notified by email.

Proactively resolve product feed issues.

The email will contain a CSV file containing clear details of what the errors are to make it quick and simple for you to resolve these. In this section, you'll find details about what each error means, and how you'll be able to resolve them.

Happy Product Feed, high performing product experts.

Being able to quickly resolve any issues helps keep your product feed up-to-date. That in turn helps you ensure your product experts have access to the latest information on pricing, availability and assortment to continue making recommendations and driving sales.

How it works

Select one or multiple assigned contacts to receive Product Feed Error Reports.

When your Product Feed is indexed, if any errors are encountered, these contacts will receive an email.

This email will contain a CSV error report, detailing exactly what the errors are. This will contain links to the relevant documentation (listed in this section) to help you quickly resolve the error:



General validation errors



Field-specific errors

Your team will be able to resolve the error(s). Following this, no further reports will be sent and the Product Feed will be indexed.

FAQs

∨ **Where can I find more information about Product Feeds and the possible errors?**

Check out the page "Integrate Product Feed" to the left to learn more about how the Product Feed works.

If you're sent an error report, we'll include the following links to documentation covering different error types and how to resolve them: [General Validation Errors](#) and [Field-specific Errors](#). You can also find this documentation in this section.

∨ **Are Product Feed Error Reports mandatory for all Virtual Shopping Merchants?**


These reports are intended to help merchants quickly resolve issues with Product Feeds and avoid resulting problems for product experts. If you would not like to receive reports, please let us know.

∨ **Who will receive the error reports?**

We'll ask you to provide details for a contact person or persons who will be notified of errors via email during the set-up process. These may be technical team members who are best placed to quickly resolve any issues.

General validation errors

Common issues encountered when processing product feeds

 **Missing/empty mandatory field**

 **Invalid URL found**

 **Invalid Number found**

 **Unable to process feed**

Missing/empty mandatory field

This field is mandatory, thus its value cannot be empty

The field in question is designated as mandatory but no value has been found for the given product.

Possible resolutions

1. Amend the product feed to provide the missing data.
2. Amend the product feed to omit products that do not meet the mandatory data requirements.
3. Amend the field definition in Klarna VS to make it optional (this comes with the tradeoff that missing data will no longer be caught for *all* products).

Invalid URL found

The following value is not a proper URL: `www.klarna.com`

Any field designated as a URL, and which contains data, *must* contain a valid URL according to the [URL Standard](#). The most common cause of this error is a missing `https://` prefix.

Possible resolutions

1. Amend the product feed to correct the invalid URLs
2. Amend the product feed to omit products for which no valid URLs are available

Invalid Number found

The following value is not a proper number: Large

Fields which must be numbers (for example, pricing fields) *must* contain valid numbers. A common cause of this error is when a number is sent with separators instead of as a plain number. For instance:

- Incorrect: 20.300,99
- Correct: 20300.99

Possible resolutions

1. Amend the product feed to correct the formatting of numbers.
2. Amend the product feed to omit products for which there is no valid numerical data

Unable to process feed

In order to index your product feed Klarna VS must be able to access the feed file, and the file must be complete and in a valid format. Common errors in this category include:

- A feed file has moved or been renamed (the product feed URL returns a 404 "Not Found" error)
- Authentication requirements for the feed URL have changed (e.g. a password has changed, or a password is now required where one was not previously required)
- A feed file is incomplete or otherwise corrupted (e.g. a truncated XML feed file which is no longer parseable as valid XML)

Possible resolutions

1. If the feed file has been moved or renamed, update your product feed URL under Account Settings in Dashboard (or provide your Klarna VS account manager with the updated details)
2. If authentication credentials have changed, provide the updated credentials to your Klarna VS account manager
 - You can identify traffic from the Klarna VS product feed indexer by its source IP addresses, which are 54.228.73.53, 54.78.118.192 and 99.81.185.60
3. If the feed file is incomplete or corrupted then you will need to check the status of the system that generates the feeds; this isn't something that Klarna VS can directly assist with
4. If you are unable to publicly host your product feed file we provide you an account on the Klarna VS SFTP server where you can upload your feed files securely; if this is something you require, speak to your Klarna VS account manager

Field-specific errors

 **SKU code**

 **Image URLs**

 **Categories**

 **Description**

SKU code

Missing SKUs

A product without ID has been found. Please revise the product feed and make sure all IDs are in place

All products in the feed *must* have a unique ID; this error occurs when the field marked as the unique ID is empty.

Possible resolutions

1. Amend the product feed to provide the missing data
2. Amend the product feed to omit products which have no unique ID

Duplicate SKUs

Klarna VS requires that products have unique identifiers. We use these unique identifiers when [tracking shopper purchase events](#), where we require that purchase events specify a unique ID that matches one we have in your product feed.

If a feed contains multiple products with the same unique ID (SKU) then it will fail validation because duplicate SKUs may have different prices or other differing data that will make our reporting inaccurate.

Possible resolutions

1. Amend the product feed to remove duplicates
2. Amend the field definition in Klarna VS to reference a different field in the product feed file (one which *does* contain unique IDs without duplicates)

Image URLs

Missing product image URLs

Klarna VS considers the product image field/s to be mandatory, so missing data is treated as a failure.

Possible resolutions

1. Amend the product feed to include the missing product images
2. Amend the product feed to omit products which do not have product images (perhaps these are test products which are not appropriate to share with shoppers)
3. Amend the product feed to include a generic placeholder image for the products that do not have actual imagery

Invalid product image URLs

In order for the product image URLs in the feed to be used by Klarna VS they must be valid URLs. See our "[Invalid URL found](#)" documentation for examples of common issues with the validation of URLs.

Categories

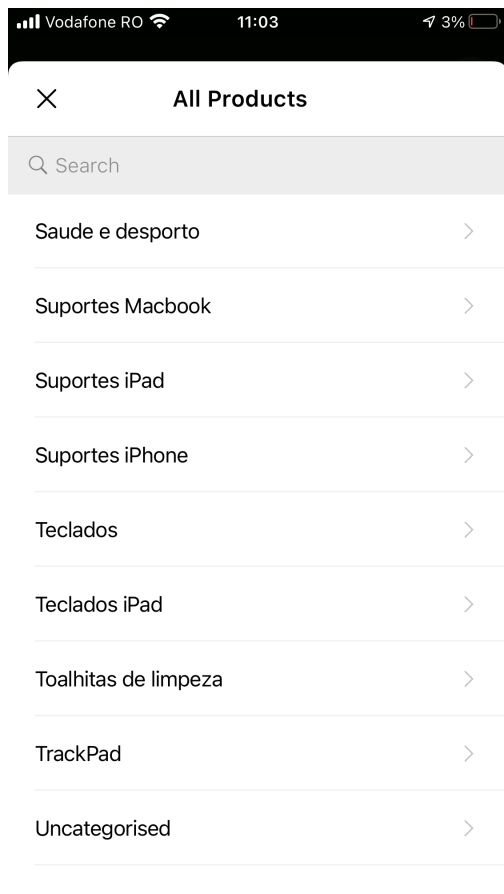
Empty category data

All products in the feed must have a category hierarchy; this is used to generate the structure of the Products feature in the Virtual Shopping Store App.

Possible resolutions

1. Amend the product feed to include the missing category data
2. Amend the product feed to omit products for which no valid category data is available
3. Amend the field definition in Klarna VS to include a default category such as "Uncategorised" (this comes with the tradeoff that missing data will no longer be caught during validation)

This is how Experts would see the `Uncategorised` category when browsing Products (it appears as a normal top-level category in the existing hierarchy):





Uncategorised

Results (1)

Search Uncategorised



Pedal de efeitos IK Multimedia iRig Stomp
44,99 €



Description

Empty description data

A common cause of validation failures with descriptions is missing/empty description data. By default Klarna VS considers the production description field to be mandatory, so missing data is treated as a failure.

Possible resolutions

1. Amend the product feed to include the missing description data
2. Amend the product feed to omit products that do not have descriptions (perhaps these are not public products, or are otherwise not suitable for recommending to shoppers)
3. Amend the field definition in Klarna VS to make it optional (this comes with the tradeoff that missing data will no longer be caught for *all* products)

Malformed description data

Product descriptions are often longer and contain a wider range of characters, than other text fields. Common problems that we see include:

1. Descriptions in a CSV product feed that contain " quotation characters or \n newlines that are not properly escaped and/or quoted
2. Incorrectly-encoded Unicode characters (for example, "" quotation characters or emoji)

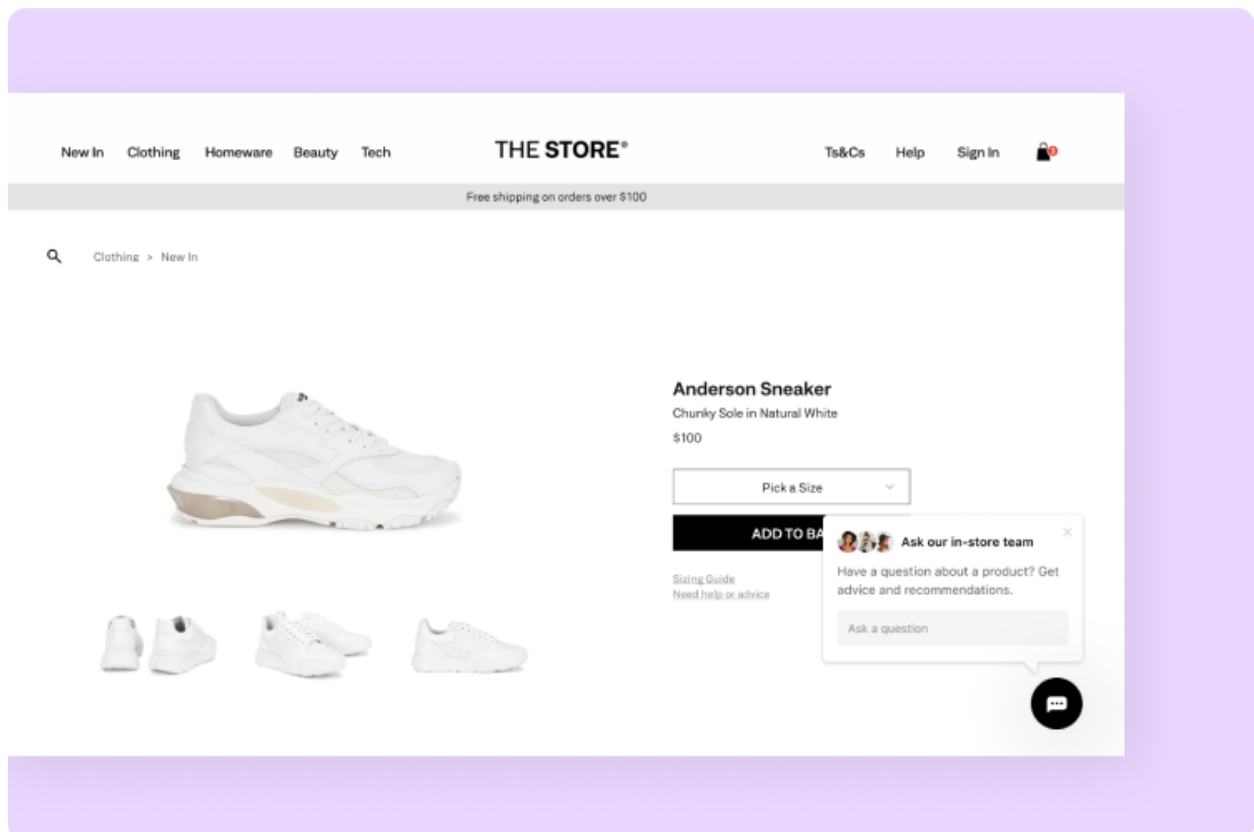
Possible resolutions

1. Amend the product feed to correctly format the description field (if using CSV format, we strongly recommend using an [RFC 4180](#) compliant CSV library rather than manually quoting/escaping values)
2. Amend the product feed to use the UTF-8 text encoding to avoid issues with Unicode characters

Set where Virtual Shopping appears

What is "Manage Display"?

Manage Display is a feature that enables you and your team to easily configure where and when Virtual Shopping On-Site Chat appears on your website. You can also see at-a-glance where Virtual Shopping is currently appearing in your website.




i Manage display is the updated version of our service previously referred to as "URL Blacklisting & Whitelisting ". As of July 2022, we've updated the name of the service, and made it easier than ever for you and your team to use.

Within the Manage Display feature, you and your team have the option to use our "simple mode", for team members of any technical ability to configure how Virtual Shopping appears, or "advanced mode", for technical teams that want to apply more complex rules for how Virtual Shopping displays.

! Advanced mode and simple mode for your display settings cannot be used together. Using simple mode will override your advanced mode settings, and vice versa.

Getting started configuring your display settings

During your launch you will be provided with a Google Sheet for you to add your preferred rules.

 When first implementing On-site Chat, the Javascript code snippet must be added to every page of your website including the checkout confirmation page, apart from sensitive checkout pages which are optional. This is important so that chats can follow shoppers from page to page and sales are tracked correctly.

While Virtual Shopping can apply rules to 90,000 URLs via Advanced Mode, we do not recommend supplying long lists of URLs because this can impact how quickly the chat loads.

Please note without either any display rules in place, the On-site Chat icon will display everywhere on the website where the tag is deployed.

When will I need to use this feature?

You will need to use this feature any time you want to adjust on which pages the On-site Chat displays. This could be during the launch of a new product, when running tests on your home page, and more.

Using Simple Mode

Simple mode is our most user-friendly option for managing your display settings, complete with clear guidance and automatic corrections for duplicate URLs. In the majority of cases, we would recommend that new merchants use this mode.

By default, the On-site Chat plugin will appear on every page of your site. You can choose to hide the chat on certain pages of your site with display rules.

About display rules

Managing how and where On-site Chat is displayed has two components:

- **The URL:** The URL defines the section of your site where Virtual Shopping shouldn't appear.
- **The display rule:** This rule defines optional exceptions to the rule.

In the simple mode, you are offered two different display rules that can be applied to your preferred URLs: **“Only ongoing chats displayed”** or **“No chats displayed”**.

Here you can determine whether you want a page to display the On-site Chat if the customer is still in the middle of an ongoing conversation, or if you would rather the page didn't display On-site Chat altogether. In order to apply one of these rules to the URL in question, you can add the URL to the appropriate text areas.

For example: We suggest hiding the On-site Chat on your checkout pages, your support/FAQ pages, or your careers page. The URLs for these pages should be added to the “No chats displayed” section.

In order to ensure that your pages can have the rules properly applied, please take the following steps:

1. Ensure that you are adding the full link (including “http://” or “https://”)
2. Ensure that you are adding the full URL for each page not applying any regex rules. In order to apply regex, please switch to Advanced Mode.

Using Advanced Mode

By default, On-site Chat will appear on every page of your site. You can choose to hide the plugin on certain pages of your site with "denied" rules. Each "denied" entry consists of two parts: a partial URL and a display rule. The partial URL defines the section of your site where Virtual Shopping shouldn't appear, and the display rule defines optional exceptions to the rule.

Using the advanced mode affords you the same rules that existed in "Simple Mode" that you can apply to any of the pages that you wish to add to your Allowed/Denied list. In order to use advanced mode, please download the template CSV, and fill in your urls with your preferred rule. Then simply upload the .csv file and hit save in order to ensure your changes are applied!

Rule 1: "False"

The false follow rule means that shoppers will never see the On-site Chat on any page matching that partial URL. The chat cannot follow them onto those pages, even if they have an active conversation started elsewhere on the site. This is the rule that is applied in the "No chats displayed" section on Simple Mode.

Rule 2: "In Progress"

The in-progress follow rule means that shoppers can only see the On-site Chat if they are in the midst of a chat with an Expert. Any pages matching that partial URL will hide the chat for users who are not currently engaged in a chat. This is the rule that is applied in the "Only ongoing chats displayed" section on Simple Mode.


In order to apply these rules, you can download the .csv template from the Advanced mode page, add your partial URLs, and add display rules. Once the .csv is updated, simply re-upload your new display rules to the Manage Display tool.

For more information on regular expressions, please visit our page on advanced mode: <https://docs.virtual-shopping.klarna.com/configure-hero/blacklisting-whitelisting/advanced-blacklisting-and-whitelisting-configuration>

To ensure that your pages can have the rules properly applied, please take the following steps:


1. Ensure that you are adding the full link (including "http://" or "https://")
2. Ensure that you are uploading a .csv file, and not a .xlsx file

Advanced Mode Examples

 Most Virtual Shopping integrations don't require advanced mode. Below we have provided some advanced examples and link to further detailed technical documentation on regular expressions.

To reiterate In most cases these are **not** required. If you have any concerns or require specific assistance please contact help.virtualshopping@klarna.com.

Introduction to regular expressions

 If your website has a large number of unique URLs we recommend using [regular expressions RegExps](#) to match multiple URLs using a single row / rule. That said, we also recommend using [RegExps](#) sparingly to balance the number of rows and the complexity of the [RegExp](#) to avoid running into performance issues with either.

A regular expression, or regex for short, is a pattern describing a certain amount of text.

The URLs that we've been using in blacklisting rule examples so far are an example of regular expressions. For instance, the special `$` character which matches the end of the URL.

Regular expressions are a complex topic and can be a powerful tool, for a comprehensive breakdown we would refer you to the following resources:

1. [Technical introduction to Regular Expressions](#)
2. [Mozilla Javascript Guide section on Regular Expressions](#)
3. [Regex101, an interactive tool for building & testing Regular Expressions](#)

Scheduling activation and expiry of rules

Virtual Shopping supports scheduling the activation date & time of a rule as well as their deactivation/expiry.

URL	Listing Type	Follow Rule	Timezone	Scheduled Date	Scheduled Time	Expires Date	Expiration Time
www.example.com/july-sale-2021	allow	false	Europe/Paris	2021-08-01	00:00		
www.example.com/2022-collection	deny	false	Europe/Paris			2022-01-01	09:00

In the above examples, we have a deny rule for our example July Sale page which activates at midnight on the 1st of August, stopping Virtual Shopping from appearing any further.

Additionally, we have a rule for our upcoming new year's collection which is **currently** blacklisted and the rule is set to expire on January 1st, from then on we would like the Virtual Shopping On-site Chat plugin will start appearing on the collection page.

Scheduling rules for an operational launch

If needed, Virtual Shopping supports scheduling your blacklisting rules for your initial operational launch. For example, the Virtual Shopping JavaScript can be added to your production site ahead of the operational launch, and set with a blacklisting rule that prevents Virtual Shopping from appearing on any page. This rule can then be set to expire at your designated go-live time.

URL	Listing Type	Follow Rule	Timezone	Scheduled Date	Scheduled Time	Expires Date	Expiration Time
.*	blacklist	false	Europe/Paris			2021-10-18	09:00
www.example.com/\$	blacklist	inprogress	Europe/Paris	2021-10-18	09:00		
www.example.com/check-out	blacklist	false	Europe/Paris	2021-10-18	09:00		
www.example.com/order-confirmation	blacklist	inprogress	Europe/Paris	2021-10-18	09:00		

In the above example we use the pattern `.*` to blacklist all possible URLs with a rule set to *expire* at 9am on the 18th of October (our operational go-live date). Our desired blacklisting rules are then scheduled to **activate** at the same time. This will prevent the On-site Chat plugin from appearing at all on our site before the allotted launch time.

Whitelisting

Although it is not commonly needed, On-site Chat does support whitelisting as an alternative to blacklisting.

With blacklisting, if a URL pattern is blacklisted then **all** other URLs are implicitly "whitelisted" (i.e. On-site Chat will appear). The opposite is true for whitelisting: **if a URL pattern is whitelisted then all other URLs are implicitly blacklisted** (note though by default they will be blacklisted with the follow rule of `inprogress`, meaning On-site Chat **will** appear if the shopper is in an active chat).

The most common use for this is to whitelist a single URL (typically a low-traffic page) to perform initial pre-launch testing in production. For further information on testing your Virtual Shopping integration, please see [our testing documentation](#).

For instance, we can whitelist our terms & conditions legal page so that the On-site Chat plugin appears but will be hidden on all other pages:

URL	Follow Rule	Type
<code>www.example.com/legal/terms\$</code>	<code>inprogress</code>	whitelist

Whitelisting can also be useful if you want On-site Chat to appear on a minority of your website's pages where using a few whitelisting rules results in a smaller/simpler overall configuration (i.e. provide few whitelisting rules instead of many blacklisting rules).

Let's imagine we only want On-site Chat to appear in two product categories, and have many other categories and non-product site sections where we do **not** want On-site Chat to appear.

URL	Follow Rule	Type
<code>www.example.com/products/outdoor-furniture</code>	<code>inprogress</code>	whitelist
<code>www.example.com/products/kitchen</code>	<code>inprogress</code>	whitelist
<code>www.example.com/checkout</code>	<code>false</code>	blacklist
<code>www.example.com/support</code>	<code>false</code>	blacklist
<code>www.example.com/careers</code>	<code>false</code>	blacklist

In the above example all product pages under the `products/outdoor-furniture` and `products/kitchen` categories will be whitelisted (On-site Chat will appear) and all other pages on the site will be blacklisted with the `inprogress` follow rule (On-site Chat won't appear unless the shopper is currently in an active chat).

We also set specific blacklist rules with the `false` follow rule to prevent On-site Chat from ever appearing on a few specific pages such as our checkout path and all support pages. We don't need to provide blacklist rules for all of our other product categories or non-product sections of the site.

Advanced examples

Language or region-specific URLs

There are many different ways that languages and geographic regions are represented in website URL paths, here we will look at some of the most common.

Showing Virtual Shopping on only one language version of a site

The first component of our URL path is a language identifier such as "fr" or "de", and we only wish to allow Virtual Shopping on the French pages of our site.

URL Pattern	Follow Rule	Type
<code>www.example.com/(?!fr FR)</code>	<code>false</code>	blacklist

The example above uses a negative lookahead regular expression to match any URL that does **not** have `fr` as its first component. [See this example in Regex101.](#)

Showing Virtual Shopping on all language versions of a site

Our site may have many localisations and we need to write blacklist rules that apply to all of them, targeting a section of our site while ignoring the language/region identifier.

In this example, we have multiple languages available on our site at URLs like `example.com/en`, `example.com/fr`, `example.com/ro` etc.

URL Pattern	Follow Rule	Type
<code>www.example.comV[a-zA-Z]{2}Vsupport</code>	<code>false</code>	blacklist
<code>www.example.comV[a-zA-Z]{2}Vcheckout</code>	<code>false</code>	blacklist

In the above example, the On-site Chat plugin will never appear on the support or checkout pages of *any* language. [See this example in Regex101.](#)

Alternatively, our site may have combined both regional and language identifiers with URLs such as `example.com/ch/ch-FR`, `example.com/ch/ch-DE` or `example.com/ca/en`

URL Pattern	Follow Rule	Type
example.com\{a-zA-Z}{2}\{a-zA-Z}{2}(-[a-zA-Z]{2})? \products\lighting	false	blacklist

In the above example we have blacklisted the `products/lighting` category for all regions & languages with a single rule. [See this example in Regex101.](#)

Allowing optional characters at the end of URLs

Many sites have pages that can also be accessed with a trailing `/` character on the end of the URL. E.g. accessing the homepage with either `example.com` or `example.com/`. If a site behaves like this and we wish to blacklist Virtual Shopping on our homepage, then we will need the following rule:

URL Pattern	Follow Rule	Type
www.example.com\?/\$	inprogress	blacklist

The above rule will blacklist Virtual Shopping on the homepage, ignoring the trailing `/` character if it's present. [See this example in Regex101.](#)

Allowing for sites to be accessible with and without the www subdomain

Some sites are served from both the `www` subdomain and the main domain itself. For instance, `www.example.com/support` and `example.com/support` may both be valid URLs serving the same content. In this case, we can omit the `www` from the pattern and the pattern will still match both versions of the URL:

URL Pattern	Follow Rule	Type
example.com\support	false	blacklist

[See this example in Regex101.](#)

Extend Virtual Shopping

Overview

Extend Virtual Shopping beyond your core implementation

Virtual Shopping On-site Chat is extendible - which means you can listen to and interact with it using the Virtual Shopping Client API. Once you have finished your core set up, it is possible to go beyond your core implementation and set up advanced events to:

- ✔ **Get deeper insights** into how shoppers are interacting with On-site Chat
- ✔ **Control how On-site Chat behaves** such as for routing chats
- ✔ **Customise how On-site Chat displays** on your website



Interaction Event Listeners



Interaction Event Methods



Update customer details



Department Filtering



Customise On-site Chat



Adjust On-site Chat position



Creating a custom chat menu

Interaction Event Listeners

An Interaction Event is an event that allows you to understand what Virtual Shopping is doing, so that you can react accordingly. You can write listeners to create custom journeys based on what Virtual Shopping is doing.

Virtual Shopping is Ready

Virtual Shopping is visible to the shopper:

```
hero("onReady", function(event) {});
```

Virtual Shopping is in Use

The shopper has a conversation in-progress:

```
hero("onReady", function(event) {  
  if (event.messengerInUse) {}  
});
```

Virtual Shopping is Not Visible

Virtual Shopping is not visible to the shopper:

```
hero("onShutdown", function() {});
```

Virtual Shopping is Shown

The Virtual Shopping On-site Chat view has been shown:

```
hero("onShow", function() {});
```

Virtual Shopping has been Hidden

The Virtual Shopping On-site Chat view has been hidden:

```
hero("onHide", function() {});
```

Example:

```
const scrollTopButton = document.querySelector("button.scroll-to-top");

hero("onReady", function() {
  scrollTopButton.style.display = "none";
});

hero("onShutdown", function() {
  scrollTopButton.style.display = "block";
});
```

Virtual Shopping Conversation has Ended

To understand when a conversation has been finished by an associate, you can use the `onConversationStatusUpdate` event:

```
hero("onConversationStatusUpdate", function(event) {
  if (event.conversationStatus === "ended") {
    console.log(event.conversationId);
    console.log(event.associateId);
    console.log(event.associateEmployeeId);
    console.log(event.now);
  }
});
```

Please note the following:

Name	Type	Description
<code>event.conversationId</code>	string	A unique ID to be able to reference the specific conversation.
<code>event.associateId</code>	string	The Virtual Shopping ID of the associate in the conversation Only populates when the shopper is using the Virtual Shopping chat and the conversation ends.
<code>event.associateEmployeeId</code>	string	The Employee ID of the associate in the conversation Only displays if this feature has been enabled as part of the Product Expert sign-up process Only populated when the shopper is using the On-site chat and the conversation ends.
<code>event.now</code>	boolean	'True' indicates that the event happened now, in real time. 'False' indicates the event happened in the past. Only populates when the shopper using the Virtual Shopping chat and the conversation ends.

As an example, to listen for a conversation which has ended whilst the shopper is still on the active page, the following can be used:

```
(event.conversationStatus === "ended" && event.now) { ... }
```

⚠ When the shopper has been down the Get Notified route there are two caveats to be aware of in relation to the Conversation Ended event:

1. If the page is closed when the chat is finished then the end of chat event will fire the next time the Virtual Shopping plugin is loaded; however the event will **not** include the associate details.
2. If the shopper is following the Get Notified link on a device different to the one where they started the original conversation then the end of chat event **won't** fire at all. For example, if the shopper starts a chat on desktop, leaves a get notified message, receives a Get Notified SMS and follows the link on their mobile device; then the end of chat event won't fire.

Conversation Rated by Shopper

The following event can be used to read the rating value and feedback contents from the rating submission by the shopper at the end of a chat. This can be used to populate in Google Analytics, for example:

```
hero("onConversationRatingSubmit", function(event) {
  console.log(event.conversationId);
  console.log(event.conversationStatus);
  console.log(event.associateId);
  console.log(event.associateEmployeeId);
  console.log(event.ratingValue);
  console.log(event.ratingFeedbackValue);
});
```

Department selected by Shopper

A shopper selects from the list of departments; note that this happens before the shopper sends their initial message to start a conversation. The event payload contains the department's name and unique ID.

```
hero("onDepartmentSelect", function(event) {
  console.log(event.departmentName);
  console.log(event.departmentId);
});
```

Name	Type	Description
event.departmentName	string	The full name of the department
event.departmentId	string	This is the same ID used in Department Filtering

Need selected by Shopper

A shopper selects from the list of needs (these are not enabled by default, please consult your Customer Success Manager for details).

```
hero("onNeedSelect", function(event) {
  console.log(event.needName);
});
```

Leave a message form is displayed

The Hero "Get Notified" form is shown to a shopper. This happens when Product Experts are not currently available and the shopper leaves a message. Note that the shopper must successfully enter a valid email or phone number for this event to be triggered.

```
hero("onGetNotifiedInit", function
```

Leave a message form is submitted

A shopper leaves a message via the "Get Notified" form.

```
hero("onGetNotifiedSubmit", function() {});
```

Shopper searches for a nearby store

A shopper submits their postcode in order to find their nearest store. This feature is not enabled by default, please consult your Customer Success Manager for details. Note that the shopper must successfully enter a valid postcode for this event to be triggered.

```
hero("onPostcodeSubmit", function() {});
```

Shopper selects a store

A shopper selects a store from the location search results. This feature is not enabled by default, please consult your Customer Success Manager for details.

```
hero("onStoreSelect", function(event) {  
  console.log(event.storeName);  
});
```

Interaction Event Methods

The Virtual Shopping Client API exposes a number of methods i.e. `show` and `hide` to allow you to interact with the On-site Chat plugin.

Showing On-site Chat

This method will show the On-site Chat window. This can only be used if Virtual Shopping is available on the page.

```
hero("show");
```

In this example, we show the On-site Chat when a button is clicked:

```
const openHeroButton = document.querySelector("button.open-hero");

openHeroButton.onclick = function() {
  hero("show");
};
```

Hiding On-site Chat

This method will hide the On-site Chat window. This will only close the chat window, if Virtual Shopping is open on the page.

```
hero("hide");
```

In this example, we can hide On-site Chat when a button is clicked:

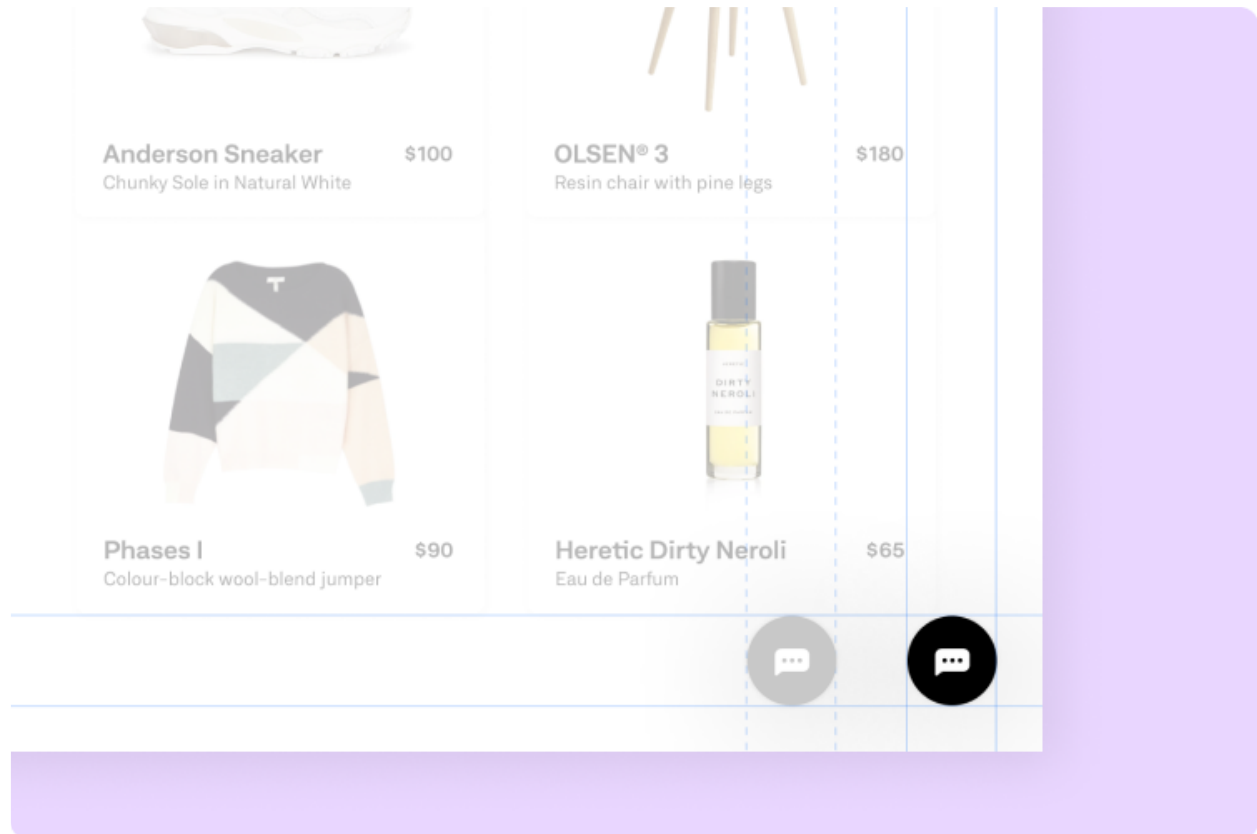
```
const closeHeroButton = document.querySelector("button.close-hero");

closeHeroButton.onclick = function() {
  hero("hide");
};
```

Adjust On-site Chat position

How to set the position of On-site Chat

Here we'll walk you through how to set precisely where you want to display Virtual Shopping on the webpage. On the pages where On-Site chat can be opened, there is a Virtual Shopping icon - a button that shoppers can click to open a chat. You can adjust its position by adding the code below to those pages.



Within the On-site Chat Tag template for Google Tag Manager, under Hero Configuration Object, two rows can be added to describe changes to the horizontal and vertical position of the icon:

```
<script>
window.HeroWebPluginSettings = {
  applicationId: "HERO-XXXX",
  translateX: "-5px",
  translateY: "40px"
};
</script>
<script>(function(i,a,m,h,e,r,o){i.HeroObject=e;i[e]=i[e]||function(){(i[e].q=i[e].q|| []).pu
```

i `translateX` and `translateY` are CSS properties that will control the Launcher Icon position using a value in pixels. For example, setting `translateX` to have a value of `-50px` will adjust the On-site Chat Launcher so that it sits 5 pixels below the default position.

i Moving the On-site Chat plugin will change the behaviour of the close/minimise function of the chat. Normally the close/minimise button is in the plugin when the chat is open, moving the plugin will move the close/minimise button to the top right corner of the chat window

Customise On-site Chat

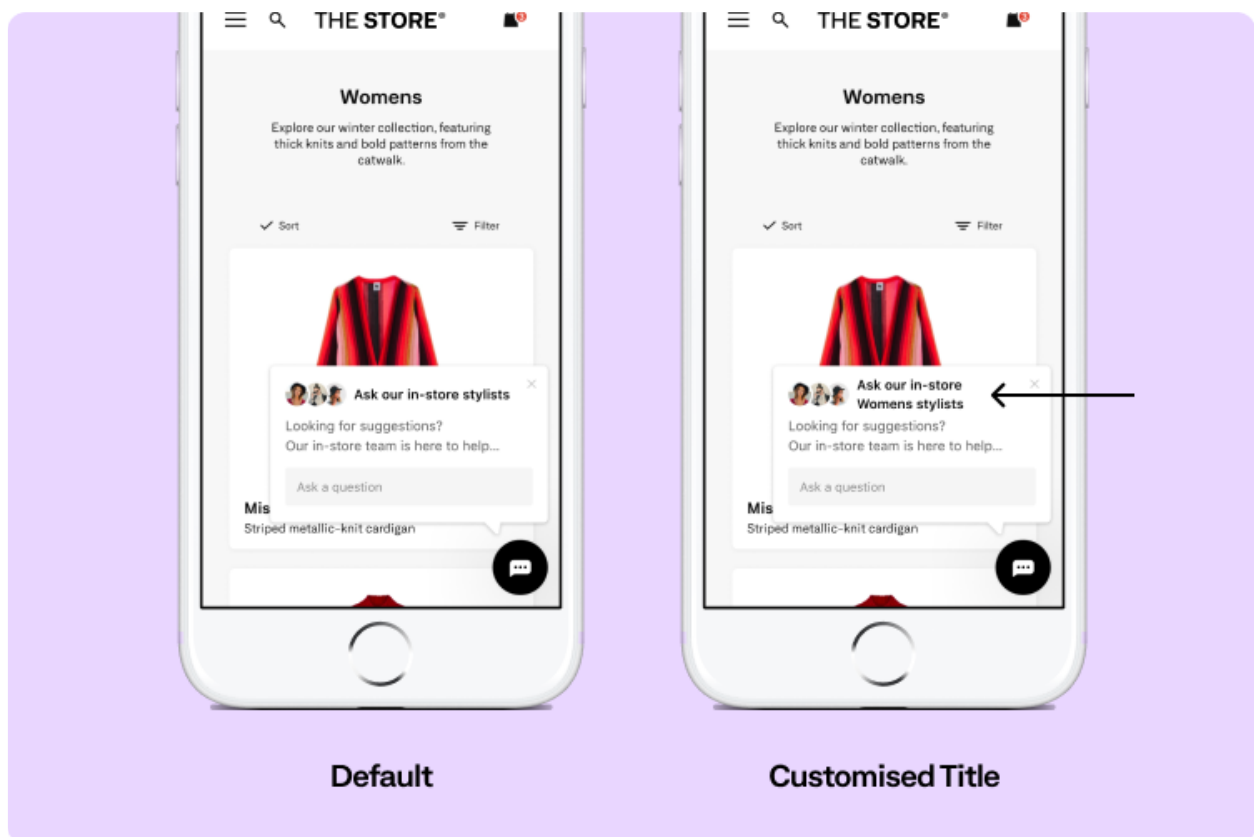
How to customise the On-site Chat title

While the wording in your On-site Chat is configurable at a site wide level with this API you can customise the title of the icon to match a specific product, category or department.

Through the Virtual Shopping API, the On-site Chat title can be updated with a specific category, subject or department name, based on the web page or section.

i This is useful when used alongside Virtual Shopping [Department Filtering](#) if particular sections or pages within a website are restricted to a single department.

The example below shows the default On-site Chat Launcher title on the left, and the updated On-site Chat Launcher title on the right, where the term `Department` has been passed through.



How to implement a different On-site Chat title

To change the On-site Chat title after the chat has initialised, you can fire an event that makes the following call:

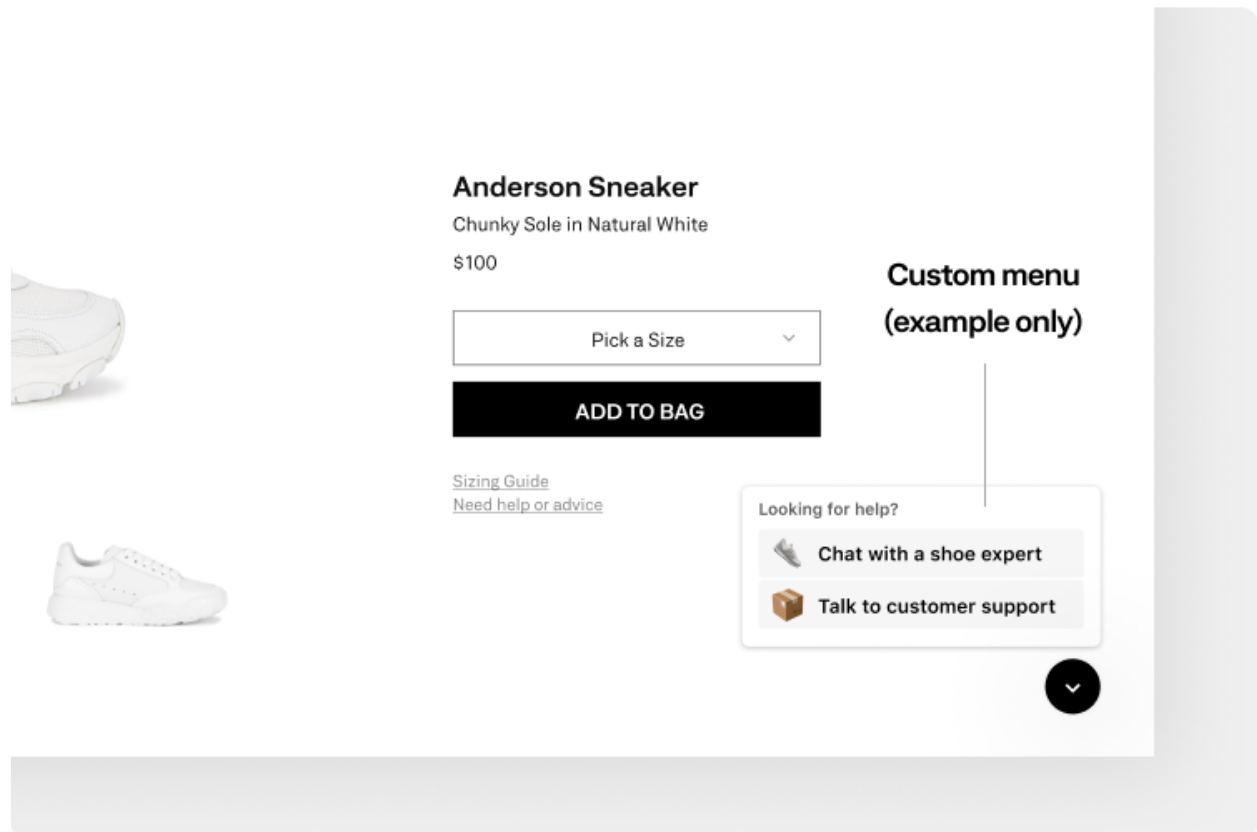
```
hero("update", { category: { title: "Department" } });
```

In this example, the term `Department` is passed through and inserted into the On-site Chat title. When using this method, the category parameter above will replace the `{ category }` part of the following title: 'Ask our in-store `{ category }` team'.


⚠ When used, this interaction event method will update across all of the Hero Launcher types which can be used: Static Launcher, Interactive Launcher and Needs Selector Launcher.

Creating a custom chat menu

Using advanced events, it is possible to customise the chat menu within the Virtual Shopping Messenger so it can sit alongside Customer Service tools. This is useful way to ensure a good shopper experience by diverting support queries away from Virtual Shopping to a Customer Service tool.



With [Interaction Event Listeners](#) and [Methods](#), you can create more complex interactions with Virtual Shopping. This guide shows how a chat menu can be introduced using Virtual Shopping interaction events.

 Please note that this an example as to how you could use a custom extension.

Hiding and unhiding the Launcher Icon

When your chat menu is in operation, the On-site Chat Launcher Icon can be hidden using CSS with the element

```
hero-launcher-container.
```

Virtual Shopping is not available

The `onShutdown` interaction event can be used to see if Virtual Shopping is not present on the page. This could be if associates are unavailable or the Follow Rules indicate that it should not show.


```
hero("onShutdown", function() {});
```

Action: Hide the option to start a chat with Virtual Shopping

Virtual Shopping is available

The `onReady` interaction event can be used to understand if Virtual Shopping is ready to be used on the page.

```
hero("onReady", function(event) {});
```

Action: Show the option to start a chat with Hero

Virtual Shopping is in use

The `onReady` interaction event can also be used to understand whether the shopper is currently in a conversation using the `messengerInUse` Boolean.

```
hero("onReady", function(event) {  
  if (event.messengerInUse) {}  
});
```

Action: Hide the chat menu altogether and show the On-site Chat icon, to benefit from the incoming notifications and alerts if the shopper is in a chat, giving them the choice to open and close when required.

Hiding / Showing Virtual Shopping

When the shopper interacts with the On-site Chat option within the chat menu, if Virtual Shopping is available, the On-site Chat can be shown using the interaction event.

```
hero("show");
```

Here is an example:

```
const openHeroButton = document.querySelector("button.open-hero");  
  
openHeroButton.onclick = function() { hero("show"); };
```

Department Filtering

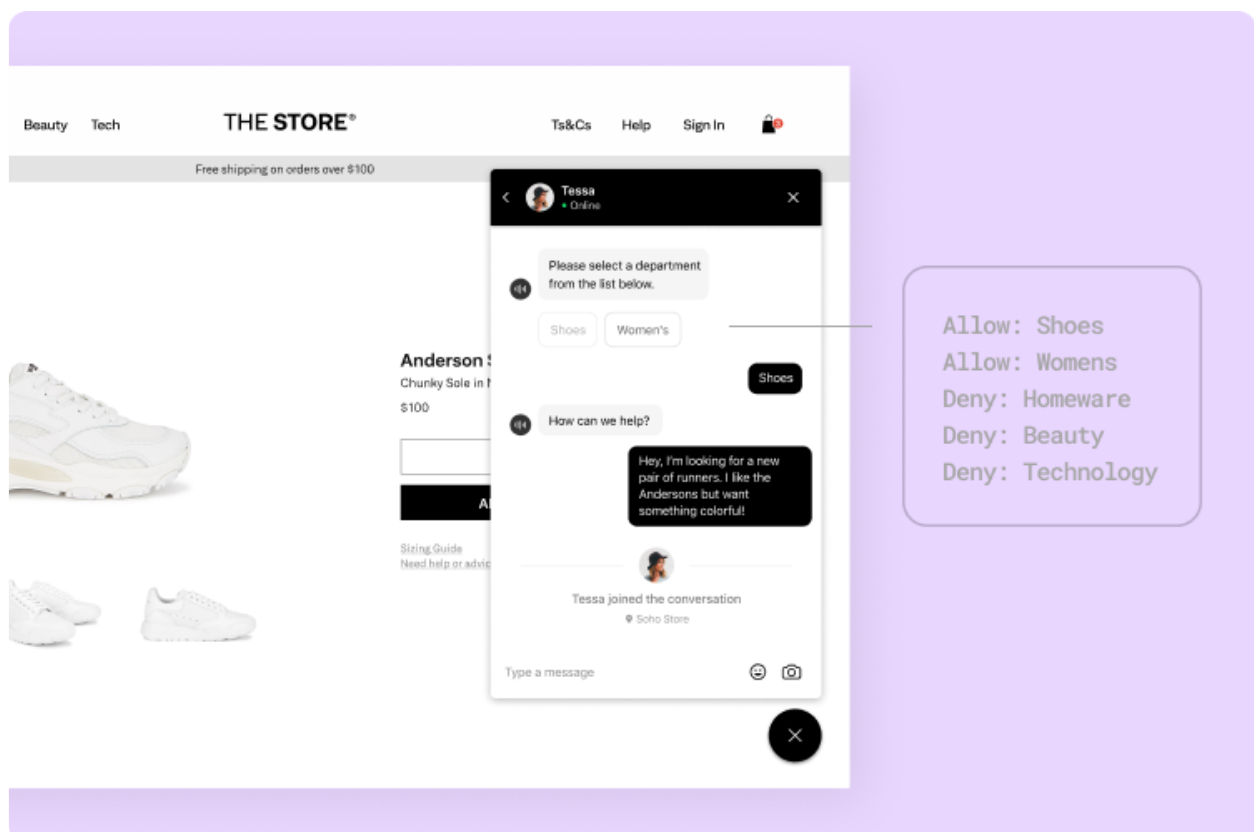
Controlling the department options available to your Shopper

By default all departments or teams configured for Virtual Shopping are shown to shoppers on all pages, allowing the shopper to select the right department for them.

However, with the Department Filtering API you are able to restrict which departments are shown to the shopper. Here are some examples of how you can use Department Filtering.

- More strictly control the journey available to your shoppers, e.g. when viewing the Women's category only allow them to start a chat with the Women's department, or when viewing a specific product only allow them to start a chat with your team who are experts in that product.
- Use availability filtering to only load a department when Associates are available in that department. Virtual Shopping's standard availability features will show the plugin if just one department is available, you can use this to hide departments on an individual level.

i When there are multiple departments that the shopper can chat with, they need to select the right one for them. However, if only one department is allowed to use Department Filtering then the shopper does not need to select a department and will go straight into starting a chat, no matter how many departments exist.



Department Filtering using Allow and Deny lists

Through the Virtual Shopping Client API, departments can be filtered using Allow and Deny lists. These are fully customisable per page so you can tailor them to the unique URL structures of your website or trigger them from events implemented by you to understand the category or product the shopper is viewing.

ⓘ To implement Department Filtering you will need to get the department IDs from Dashboard. You can find these under Settings > Account Settings > Departments. Note, department IDs are case sensitive.

If you don't see the option for Settings in the Dashboard navigation, or the option for Departments, then contact help.virtualshopping@klarna.com

Department Filtering Rules

When allowing and denying departments, it's important to understand these rules:

- ⓘ • When one department is added to the Allow list, any departments not listed will be hidden.
- When one department is added to the Deny list, any departments not listed will still be visible.
- Only an Allow or Deny need to be used on any one page, however when using Allow and Deny together, Allow will always take precedence.

Allowing Departments

ⓘ For each type of Allow list below you will see two examples, one where only a single department is allowed and one where multiple departments are allowed.

Allow a Department

Used when you only want to show certain departments on a page. Any departments not in the allow list will not be shown.

Allowing a single department:

```
window.HeroWebPluginSettings = {  
  departmentAllowList: [  
    {  
      id: "43a0B7JdY" // Department ID  
    }  
  ]  
};
```

Allowing multiple departments:

```
window.HeroWebPluginSettings = {  
  departmentAllowList: [  
    {  
      id: "43a0B7JdY" // Department ID  
    },  
    {  
      id: "27He0Lj38a" // Department ID  
    },  
    {  
      id: "HIW73901p1" // Department ID  
    }  
  ]  
};
```

Allow a Department if it is Available

You only want to show a department on a page when an Associate is available in that department.

Allowing a single department:

```
window.HeroWebPluginSettings = {  
  departmentAllowList: [  
    {  
      id: "43a0B7JdY", // Department ID  
      constraint: {  
        departmentAvailable: true  
      }  
    }  
  ]  
};
```

Allowing multiple departments:

```

window.HeroWebPluginSettings = {
  departmentAllowList: [
    {
      id: '43a0B7JdY', // Department ID
      constraint: {
        departmentAvailable: true
      }
    },
    {
      id: '27He0Lj38a', // Department ID
      constraint: {
        departmentAvailable: true
      }
    },
    {
      id: 'HIW7390lp1', // Department ID
      constraint: {
        departmentAvailable: true
      }
    }
  ]
};

```

Allow a Department if an Active Conversation Exists

You only want to show a department on a page when the shopper has a chat in progress with a department.

Allowing a single department:

```

window.HeroWebPluginSettings = {
  departmentAllowList: [
    {
      id: "43a0B7JdY", // Department ID
      constraint: {
        conversationInProgress: true
      }
    }
  ]
};

```

Allowing multiple departments:

```

window.HeroWebPluginSettings = {
  departmentAllowList: [
    {
      id: '43a0B7JdY', // Department ID
      constraint: {
        conversationInProgress: true
      }
    },
    {
      id: '27He0Lj38a', // Department ID
      constraint: {
        conversationInProgress: true
      }
    },
    {
      id: 'HIW7390lp1', // Department ID
      constraint: {
        conversationInProgress: true
      }
    }
  ]
};

```

Allow a Department if a Previous Conversation Exists

You only want to show a department on a page when a user has had a previous conversation with that department.

Allowing a single department:

```

window.HeroWebPluginSettings = {
  departmentAllowList: [
    {
      id: "43a0B7JdY", // Department ID
      constraint: {
        conversationExists: true
      }
    }
  ]
};

```


Allowing multiple departments:

```

window.HeroWebPluginSettings = {
  departmentAllowList: [
    {
      id: '43a0B7JdY', // Department ID
      constraint: {
        conversationExists: true
      }
    },
    {
      id: '27He0Lj38a', // Department ID
      constraint: {
        conversationExists: true
      }
    },
    {
      id: 'HIW73901p1', // Department ID
      constraint: {
        conversationExists: true
      }
    }
  ]
};

```

Denying Departments

 For each type of Deny list below you will see two examples, one where only a single department is denied and one where multiple departments are denied.

Deny a Department

You only want to prevent one (or more) departments from showing. All other departments will show.

Denying a single department:

```

window.HeroWebPluginSettings = {
  departmentDenyList: [
    {
      id: "43a0B7JdY" // Department ID
    }
  ]
};

```

Denying multiple departments:

```
window.HeroWebPluginSettings = {
  departmentDenyList: [
    {
      id: "43a0B7JDdY"
    },
    {
      id: "27He0Lj38a"
    },
    {
      id: "HIW7390lp1"
    }
  ]
};
```

Deny a Department if it is Unavailable

Do not show a department if a department is unavailable, or a shopper hasn't had a conversation in that department.

Denying a single department:

```
window.HeroWebPluginSettings = {
  departmentDenyList: [
    {
      id: "43a0B7JDdY", // Department ID
      constraint: {
        departmentAvailable: false
      }
    }
  ]
};
```

Denying multiple departments:


```

window.HeroWebPluginSettings = {
  departmentDenyList: [
    {
      id: '43a0B7JdY', // Department ID
      constraint: {
        departmentAvailable: false
      }
    },
    {
      id: '27He0Lj38a', // Department ID
      constraint: {
        departmentAvailable: false
      }
    },
    {
      id: 'HIW7390lp1', // Department ID
      constraint: {
        departmentAvailable: false
      }
    }
  ]
};

```

Deny a Department if Previous Conversation Does Not Exist

Deny a department from showing if a conversation does not exist. This will only show a department if a conversation has already happened with this department.

Denying a single department:

```

window.HeroWebPluginSettings = {
  departmentDenyList: [
    {
      id: "43a0B7JdY", // Department ID
      constraint: {
        conversationExists: false
      }
    }
  ]
};

```


Denying multiple departments:

```
window.HeroWebPluginSettings = {
  departmentDenyList: [
    {
      id: '43a0B7JdY', // Department ID
      constraint: {
        conversationExists: false
      }
    },
    {
      id: '27He0Lj38a', // Department ID
      constraint: {
        conversationExists: false
      }
    },
    {
      id: 'HIW7390lp1', // Department ID
      constraint: {
        conversationExists: false
      }
    }
  ]
};
```

Configurations

Attribute	Required	Type	Note
<code>id</code>	Mandatory	string	Department IDs can be requested from Virtual Shopping for Staging and Production retail applications.
<code>constraint</code>	Optional	object	
<code>constraint.departmentAvailable</code>	Optional	boolean	The department's current availability state determines if it is considered to be in the Allow / Deny list.
<code>constraint.conversationsInProgress</code>	Optional	boolean	Whether the Shopper has a conversation in progress determines if it is considered to be in the Allow / Deny list.
<code>constraint.conversationsExist</code>	Optional	boolean	Whether the Shopper has an existing conversation with the department determines if it is considered to be in the Allow / Deny list.

Suggested Checklist for Creating New Departments and Implementing Department Filtering While Virtual Shopping is Already Live on Your Site

 When a department is created through your Dashboard it will immediately show across all pages where the On-site Chat plugin appears. If this is a problem then you may want to follow the order of steps below for rolling out Department Filtering.

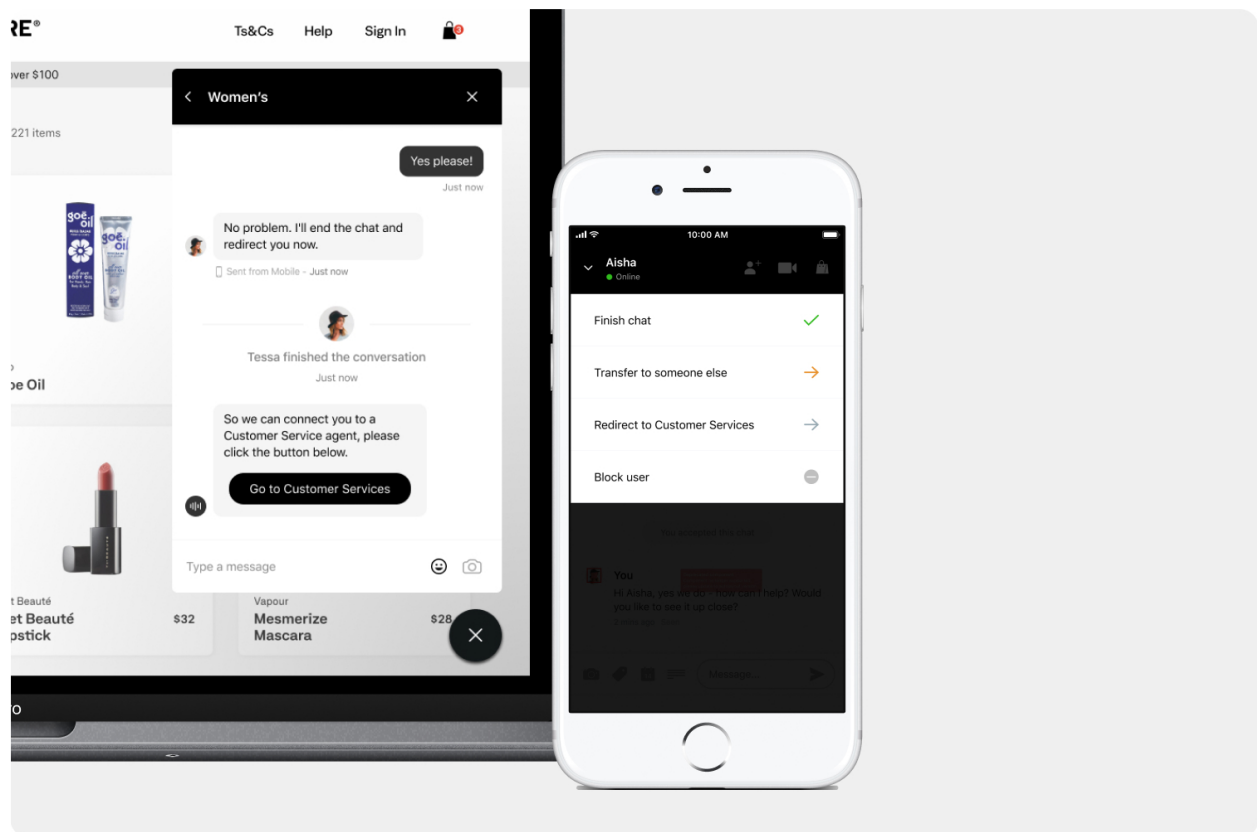
1. Implement Department Filtering for your existing departments, this will mean that the plugin continues to behave as normal while you create the new departments
2. Create new departments Settings > Account Settings > Departments > Add Department
3. Assign relevant Associates to new departments, remember Associates can be in multiple departments
4. Update Department Filtering to use the new department IDs where you want them to appear

Using Department Filtering to deactivate a department

Virtual Shopping does not currently support deleting or removing departments. If you no longer want a department to be available to your Shoppers then simply implement Department Filtering and add the relevant department to a deny list on all pages.

Redirect to Customer Service


Sending shoppers to the right place




Sometimes, shoppers will start a Virtual Shopping chat when they actually want to speak to your Customer Service team. It's easy to redirect these shoppers before or during a Virtual Shopping chat so that they always reach the right person and your team members only receive relevant chats.

You can redirect shoppers to your Customer Service in 3 ways:

Before a chat:

 Redirect before a chat with Filtering

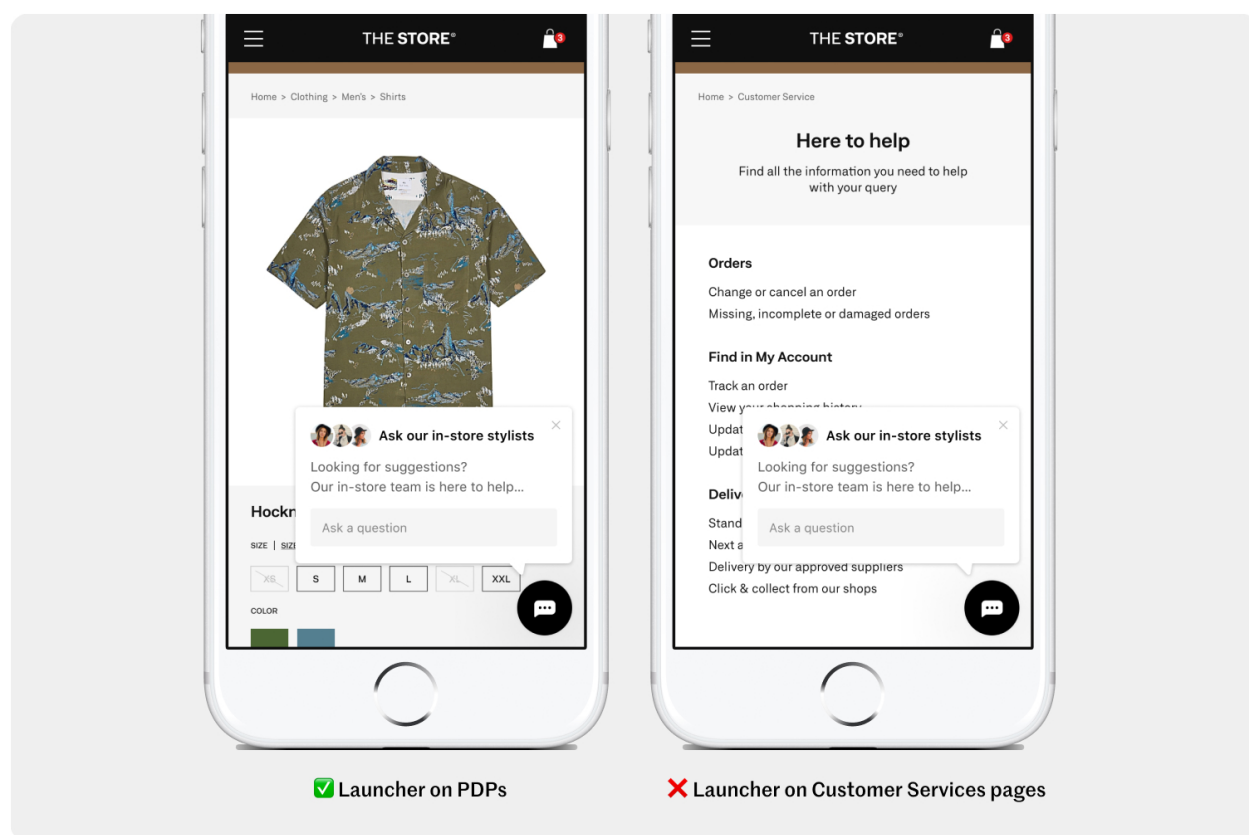
 Redirect before a chat with the Needs Selector

During a chat:

 Redirecting during a chat

Before we take a look at these opportunities, it's first important to place On-site Chat correctly on your website in relation to your Customer Service pages or tool.

Placing On-site Chat in relation to Customer Service pages



On-site Chat should appear on pages that are most likely to drive sales queries, usually category and product-level pages. As a result, it is **not** normally suitable to place On-site Chat on your Customer Service pages or home page.

- ✓ To ensure Hero does not appear on your homepage, we recommend you first [deploy Hero across your whole website](#) and then use [URL blacklisting and whitelisting](#) to ensure it is not showing on the home page.

Placing On-site Chat in relation to a Customer Service tool

If you are using another tool on your website dedicated to handling Customer Service chats, you will want to ensure that it is not showing at the same time as the Virtual Shopping On-site Chat. You can do this using [Interaction Events](#)

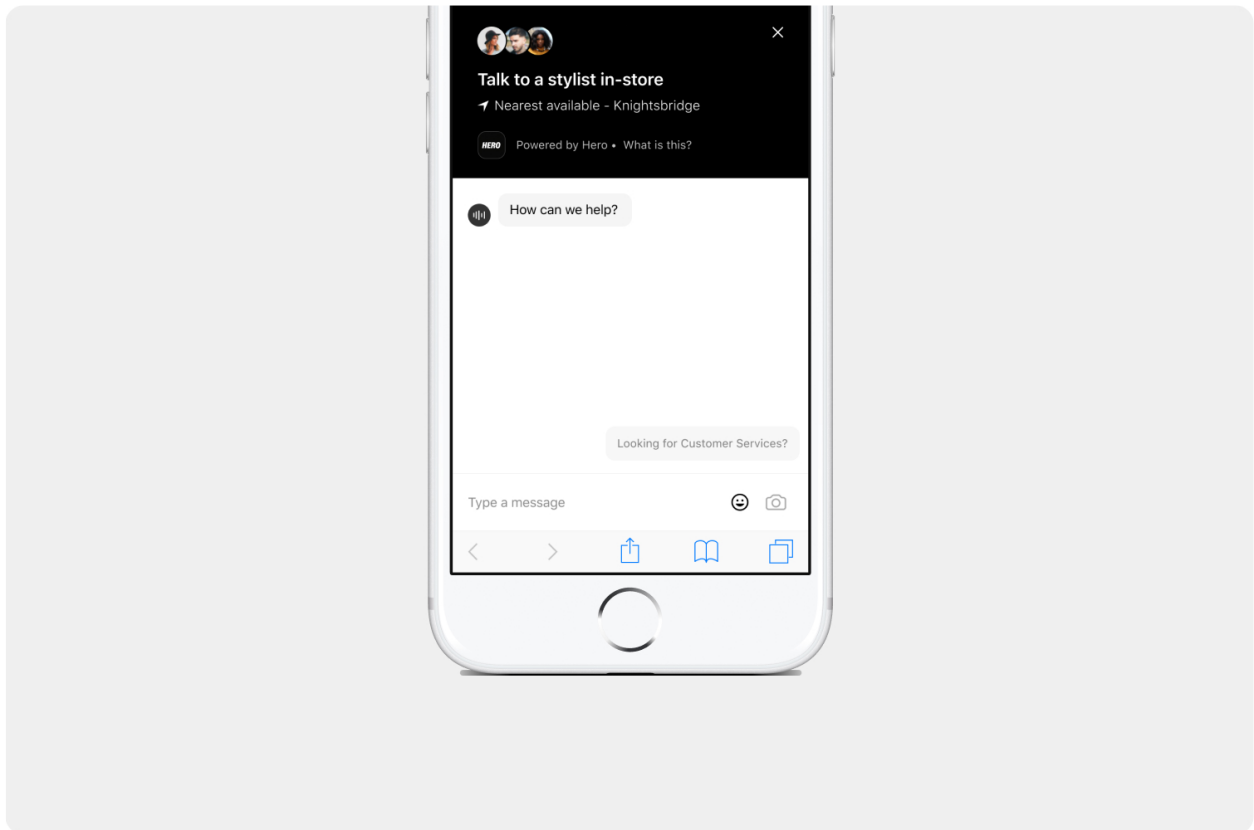
Every time a page loads, On-site Chat fires an event to the page to say if it will load and what state it will be in. For example, it might say it's going to load with the On-site Chat open.

With a [simple listener script](#) you can understand if Virtual Shopping is loading to the page and decide whether or not to load your Customer Service chat solution instead. To trigger your Customer Service tool in this way, you can use an Interaction Event Listener. Virtual Shopping offers some out-the-box scripts as examples, or you can build one yourself.

Redirect before a chat with Filtering

Using Customer Service Filtering

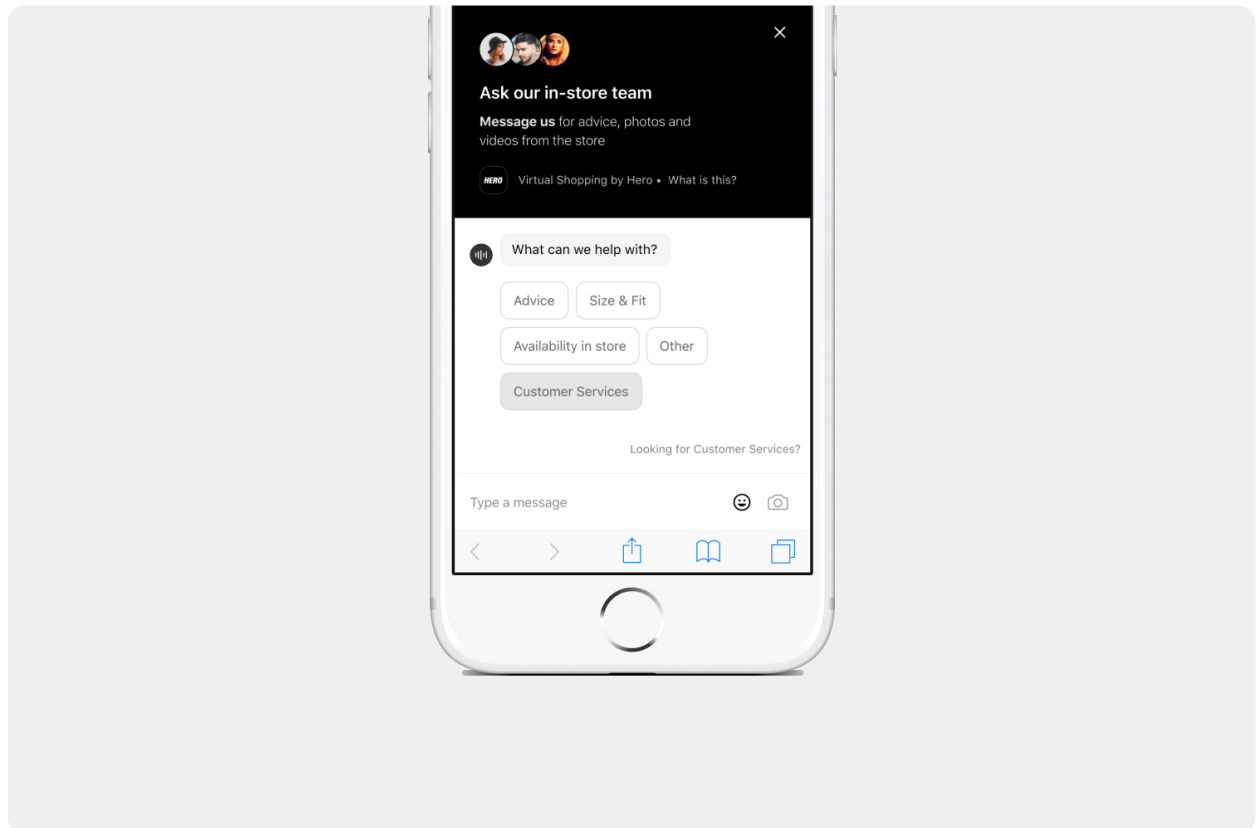
If a shopper has opened the Virtual Shopping Messenger, there is signposting to guide them to your existing Customer Service page. This is a standard part of the Virtual Shopping setup process and you don't need to implement anything further.



Redirect before a chat with the Needs Selector

Using the Needs Selector 🙌

You can give shoppers the option to select the nature of their query before they start a chat using using the Needs Selector. If the shopper selects the Customer Service category then they will be redirected to another page on your website, or to another tool.



When a shopper chooses this option, Virtual Shopping then redirects them to your Customer Service page. If you have another tool you would like to direct the shopper to then you can set up a basic listener script to hide On-site Chat and open the other tool to seamlessly put your customer into your preferred channel.

i This option requires the Needs Selector to be active and the Customer Services option needs to be turned on. If you are interested in setting up the Needs Selector in Hero Messenger, please get in touch with your Customer Success Manager.

Virtual Shopping offers a number of [example out-the-box listener scripts](#) for Customer Service tools or you can create your own.

Breakdown of steps

Follow these simple steps to enable this customer journey:

1. Add the `onCustomerServiceRedirect` listener script to your website

Here is an example of how to implement the listener script:

```
hero("onCustomerServiceRedirect", function (event) {  
    var customerServiceUrl = event.customerServiceUrl;  
    window.open(customerServiceUrl, '_blank');  
});
```

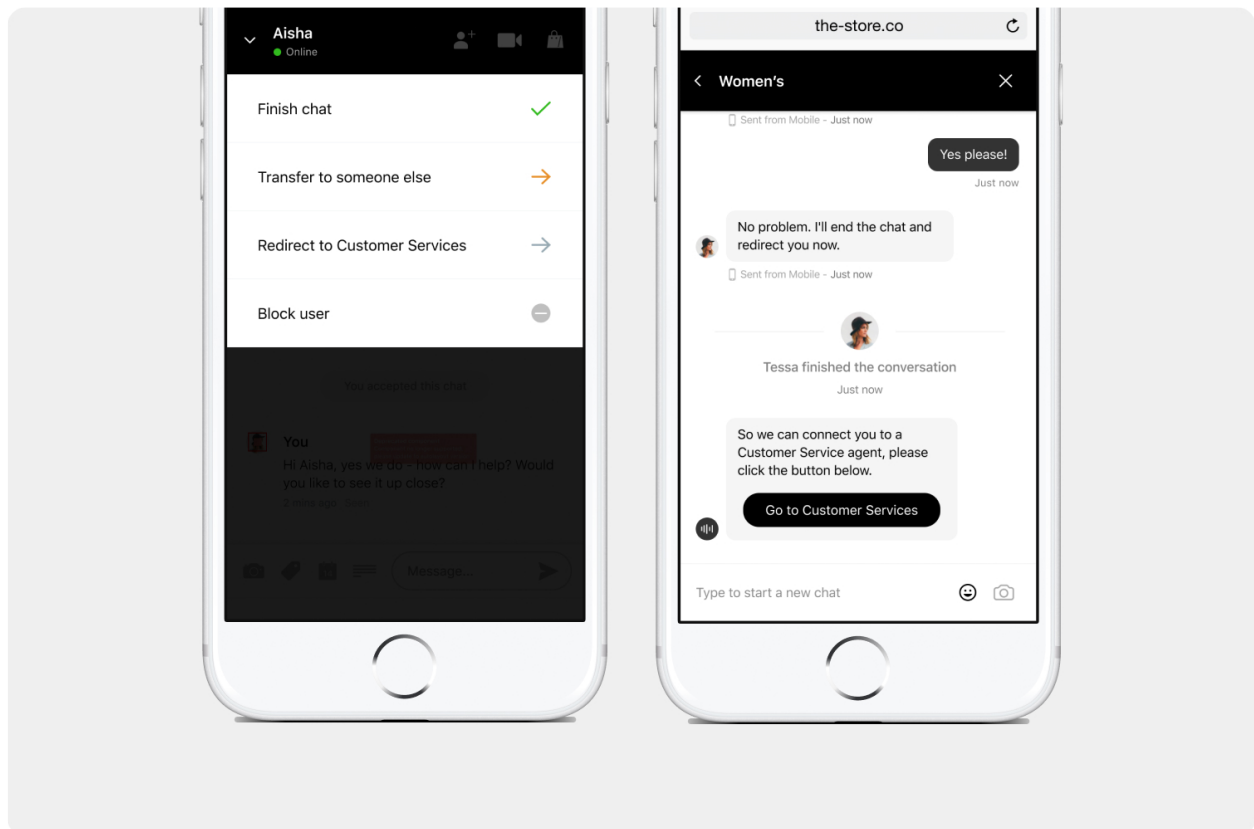
2. Let the Klarna team know that you have implemented the listener script. They will then switch on the Needs Selector in Virtual Shopping Messenger so that shoppers can choose to be redirected to your Customer Service tool or Customer Service page

3. Test the implementation. We recommend you test that the integration is behaving as expected on Staging before pushing to Production

Redirecting during a chat

Redirecting shoppers when in a chat

If a shopper has already started a chat with a member of your team but would be better served by Customer Service, team members can transfer that shopper to Customer Service during a chat.



After the team member taps on 'Redirect to Customer Services' button in the app there are two points in the journey where you can complete the switchover. Choose whichever gives the shopper experience you want. The solution is the same, it just relies on listening to a different on page event. With a simple listener script you can detect the event, close Virtual Shopping On-site Chat and open your Customer Service tool.

1. As soon as the transfer journey is initiated an event called `onCustomerServiceInstantRedirect` will fire to the page.
2. After this a clear call to action for the shopper to chat to Customer Services is sent into the chat; when clicked this fires an event called `onCustomerServiceRedirect`

This means you can instantly switch tools, or give your shopper the choice to select to talk to Customer Service.

Virtual Shopping offers a number of [example out-the-box listener scripts](#) for Customer Service tools or you can create your own.

Breakdown of steps

Follow these simple steps to enable this customer journey:

1. Add the `onCustomerServiceRedirect` listener script to your website

Here is an example of how to implement the listener script:

```
hero("onCustomerServiceRedirect", function (event) {
  var customerServiceUrl = event.customerServiceUrl;
  window.open(customerServiceUrl, '_blank');
});
```

2. Let the Klarna Virtual Shopping team know that you have implemented the the Listener Event script. They will then switch on the transfer feature so your teams can send the redirect button into the chat
3. Test the implementation. We recommend you then test the integration is behaving as expected on Staging before pushing to Production

Sharing the conversation history with a customer service agent

If you would like to share the conversation with a customer service agent, we provide a transcript property containing the conversation history. The transcript property can be accessed from the event object of the `onCustomerServiceRedirect` and `onCustomerServiceInstantRedirect` events.

Here is an example of how this can be implemented. In this example, the third-party customer service tool is initialised, and the transcript is sent as the first message from the shopper:

```
hero("onCustomerServiceRedirect", function (event) {
  customerServiceSDK.init();
  customerServiceSDK.sendMessage(event.transcript);
});
```

The transcript value is a string, and an example value is shown below:

```
Shopper [11:13, 11th Aug 22]
Hi, I need help with a return

Expert [11:14, 11th Aug 22]
Hi there, Our customer service team can help with that. I'll transfer you!
```

Example Listener Scripts

Integration scripts

Using the Virtual Shopping Client API, you can set up a redirect to any Customer Service tool. Virtual Shopping gives you the means to create your own integration or use an out-the-box integration with:

- Kustomer
- iAdvize
- RingCentral

Custom integration scripts

If you would like to create your own integration, you can do so using the Virtual Shopping Client API; create a Customer Service redirect listener to open your own Customer Service tool when this event is fired.

The name for this event is `onCustomerServiceRedirect` and an example integration would look like this:

```
hero("onCustomerServiceRedirect", function (event) {
  var customerServiceUrl = event.customerServiceUrl;
  window.open(customerServiceUrl, '_blank');
});
```

Out-the-box Example scripts

This option means you don't need to integrate the `onCustomerServiceRedirect` listener script. Instead, you can just load an additional script to the page.

We suggest you load scripts in this order:

1. Customer Service Redirect Script (a list of example scripts follows below)
2. Customer Service tool script
3. Virtual Shopping script

Kustomer

Here is an example of a custom redirect script for Kustomer:

```
hero('onCustomerServiceRedirect', function (a) {  
  // 1. add the kustomer script to the page  
  !function(a,b,c,d){a.Kustomer=c,c._q=[],c._i=[],c.init=function(a){function b(a,b){a[b]=fu  
  
  // 2. initialise the kustomer chat client  
  Kustomer.init('YOUR_API_KEY');  
  
  // 3. start and open the kustomer chat client and hide hero  
  Kustomer.start({ icon: !1 }, function () {  
    Kustomer.open();  
    hero('hide');  
  });  
});
```

iAdvize

Run this to see an example of a custom redirect script for iAdvize:

```

(function () {
  const ONE_SECOND = 1000;
  const THIRTY_SECONDS = ONE_SECOND * 30;

  let interval;
  let heroIsReady = false;
  let areAnyHeroConvInProgress = false;

  const isElementVisible = (element) => {
    if (element) {
      return window.getComputedStyle(element).display !== 'none';
    }
    return false;
  };

  const hideByElementSelector = (selector) => {
    const css = `${selector} { visibility: hidden !important; }`;
    const head = document.head || document.getElementsByTagName('head')[0];
    const style = document.createElement('style');
    head.appendChild(style);
    style.type = 'text/css';
    style.setAttribute('data-hero-override', '');
    style.appendChild(document.createTextNode(css));
  };

  const hideByElementSelectors = (selectors) => {
    selectors.forEach(hideByElementSelector);
  };

  const getIadvizeStaticButton = () => {
    const iadvizeStaticButton = document.querySelector('#idz_btn #idz_fonline');
    if (iadvizeStaticButton) {
      return iadvizeStaticButton;
    } else {
      const iframe = document.querySelector('iframe[id*="iframe-notification"]');
      return iframe && iframe.contentDocument.querySelector('button[class*="NotificationButt');
    }
  };

  const iadvizeShow = () => {
    const styles = document.head.querySelectorAll('style[data-hero-override]');
    styles.forEach((style) => {
      if (style) {
        style.remove();
      }
    });
  };

  const iadvizeHide = () => {
    hideByElementSelectors([
      `div[id^='idz']`, `div[id*=' idz']`,
      `iframe[id^='iframe-notification']`, `iframe[id*=' iframe-notification']`,
    ]);
  };
}

```

```

const iadvizeOpen = () => {
  const iadvizeStaticButton = getIadvizeStaticButton();
  if (iadvizeStaticButton) {
    iadvizeStaticButton.click();
  }
};

const heroShow = () => {
  const heroContainer = document.querySelector('#hero-iframe-container');
  heroContainer.style.visibility = 'visible';
};

const heroHide = () => {
  const heroContainer = document.querySelector('#hero-iframe-container');
  heroContainer.style.visibility = 'hidden';
};

const heroShutdown = () => {
  heroHide();
  window.hero('shutdown');
  clearInterval(interval);
};

const getIadvizeElements = () => {
  const iadvizeStaticButton = getIadvizeStaticButton();
  const iadvizeActiveButton = document.querySelector('#idz_chatbar_mini');
  const iadvizeChatWindow = document.querySelector('#idz_chatglobal');
  return [iadvizeStaticButton, iadvizeActiveButton, iadvizeChatWindow];
};

const getIadvizeStatus = () => {
  const [iadvizeStaticButton, iadvizeActiveButton, iadvizeChatWindow] = getIadvizeElements();
  const iadvizeInStaticState = isElementVisible(iadvizeStaticButton);
  const iadvizeConvIsHidden = isElementVisible(iadvizeActiveButton);
  const iadvizeConvIsVisible = isElementVisible(iadvizeChatWindow);
  if (iadvizeInStaticState) {
    return 'static';
  } else if (iadvizeConvIsHidden || iadvizeConvIsVisible) {
    return 'active';
  } else {
    return null;
  }
};

const processStatus = () => {
  const iadvizeStatus = getIadvizeStatus();
  const iadvizeIsOnThePage = !!iadvizeStatus;
  if (iadvizeIsOnThePage && heroIsReady) {
    if (iadvizeStatus === 'active' && !areAnyHeroConvInProgress) {
      iadvizeShow();
      heroShutdown();
    }
    clearInterval(interval);
  }
};

```

```

/**
 * Step 1. Hide both iadvize and the Hero messenger from the shopper.
 */
iadvizeHide();
heroHide();

/**
 * Step 2. Check the status of iadvize every second. If iadvize is actively
 * being used, show iadvize and shutdown the Hero messenger.
 */
processStatus();
interval = setInterval(processStatus, ONE_SECOND);

/**
 * Step 3. Listen to Hero onReady event. If this event is fired, show the Hero
 * messenger whilst continuing to check the status of iadvize every second
 * (just in case iadvize is actively being used).
 */
window.hero('onReady', ({ messengerInUse }) => {
  heroIsReady = true;
  areAnyHeroConvInProgress = messengerInUse;
  heroShow();
  processStatus();
});

/**
 * Step 4. Listen to Hero onCustomerServiceRedirect event. If this event is
 * fired, show iadvize, open iadvize and shutdown the Hero messenger.
 */
window.hero('onCustomerServiceRedirect', ({ customerServiceUrl }) => {
  const iadvizeStatus = getIadvizeStatus();
  const iadvizeIsOnThePage = !!iadvizeStatus;
  if (iadvizeIsOnThePage) {
    iadvizeShow();
    iadvizeOpen();
    heroShutdown();
  } else {
    window.open(customerServiceUrl, '_blank');
  }
});

/**
 * Step 5. Stop checking the status of iadvize every second after 30 seconds
 * pass.
 */
setTimeout(() => {
  clearInterval(interval);
}, THIRTY_SECONDS);
})();

```

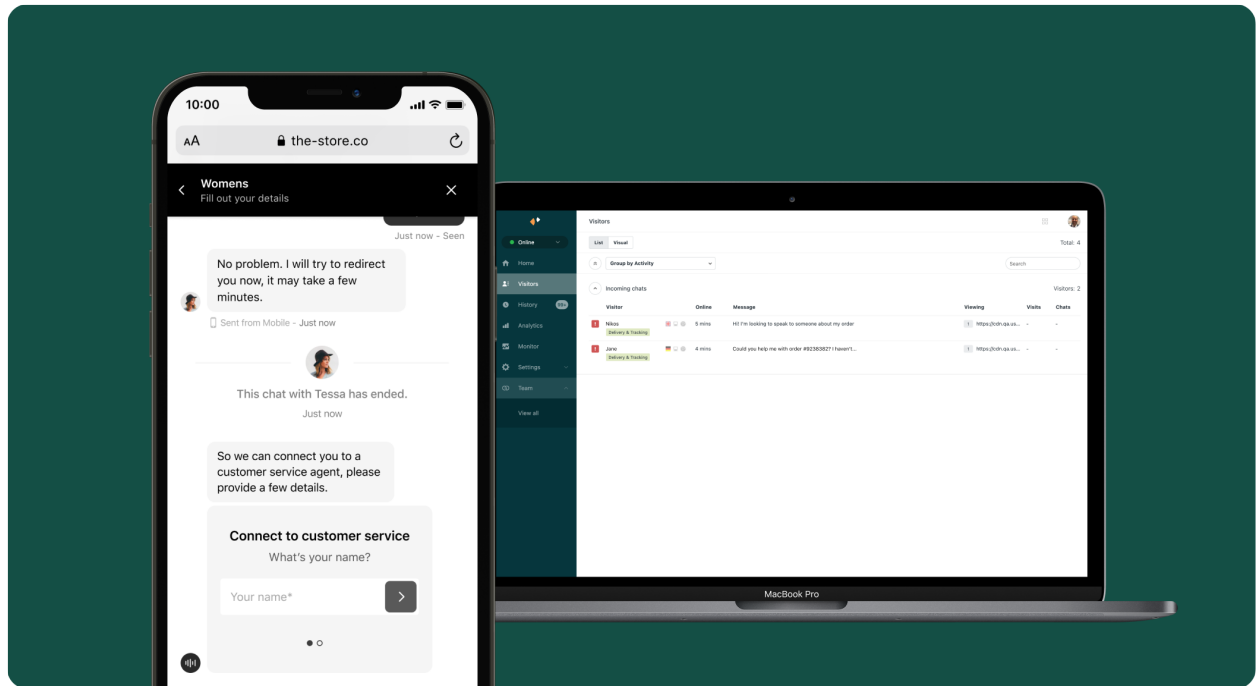
RingCentral

Run this to see an example of a custom redirect script for RingCentral:


```
<script async type="text/javascript" src="https://cdn.usehero.com/ringcentral.js"></script>
```

Integrate Zendesk

Simple steps to connect Zendesk and Virtual Shopping.



Sending shoppers to the right place

Using our Virtual Shopping integration with Zendesk, product experts can easily transfer shoppers to Customer Service teams, without the shopper needing to leave the Messenger.

⚠ Please note: the Zendesk and Virtual Shopping integration is only available if you have a Zendesk Enterprise licence.

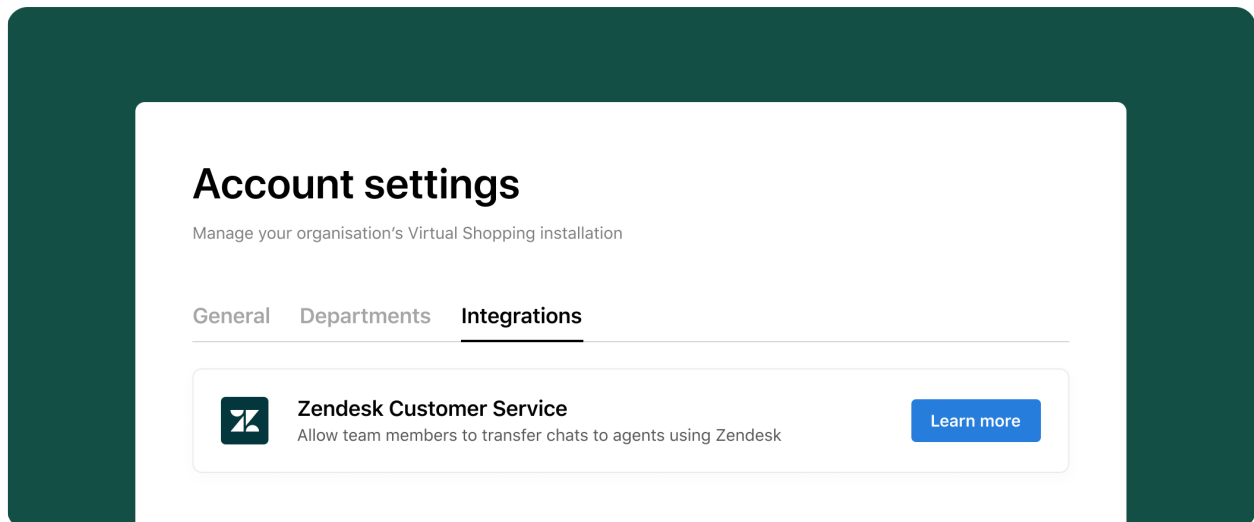
In this article, we'll walk through the simple steps to integrate Zendesk with Virtual Shopping following an activation flow in the Dashboard. To get set up, follow the steps below:

Step 1: Before you begin

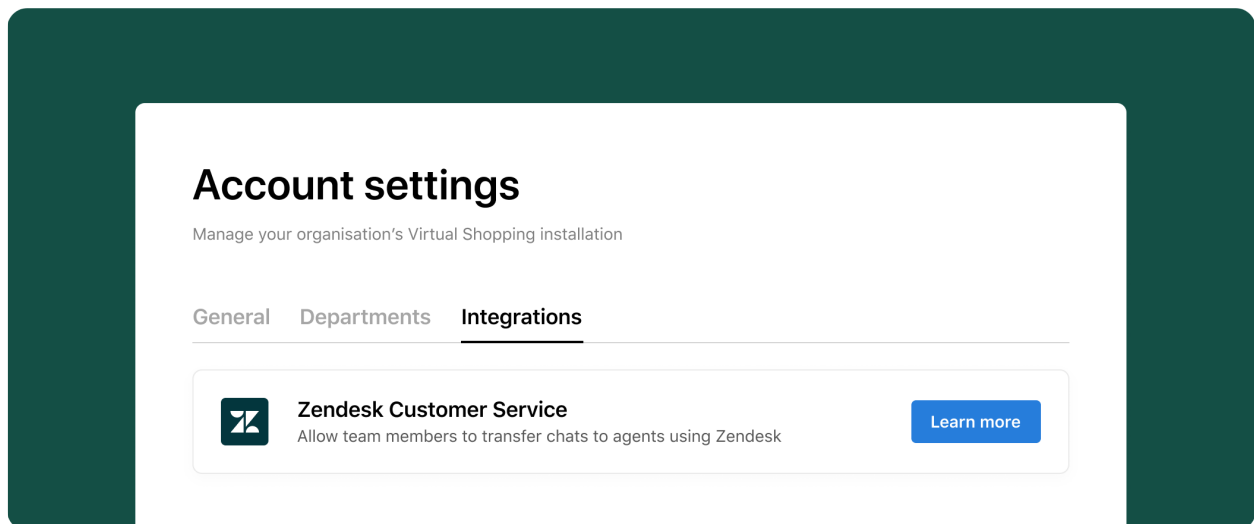
1. Before you start the setup flow, please ensure:

- The user with admin permissions for Settings is logged into the Dashboard
- Ensure you've been approved for the Zendesk Integration. Simply send an email to help.virtualshopping@klarna.com including your name, role, and company. In this email, please specify if we should switch on the in-app transfer menu, the Needs Selector or both.
- Your Partnership Success Manager is aware that you are interested in the Zendesk integration
- If you have a Content Security policy in place, we recommend you put in place the directives set out by Zendesk [here](#).
- Chat triggers are switched off. This is because these messages may interfere with communications we have created within the integration. We recommend at a minimum, you disable the 'Chat Rescuer' chat trigger type. To disable a chat trigger from the Zendesk Chat Dashboard, read [this article](#).
- If you are already using our 'onCustomerServiceRedirect' listener script from your website you should remove this once the Zendesk integration is activated.

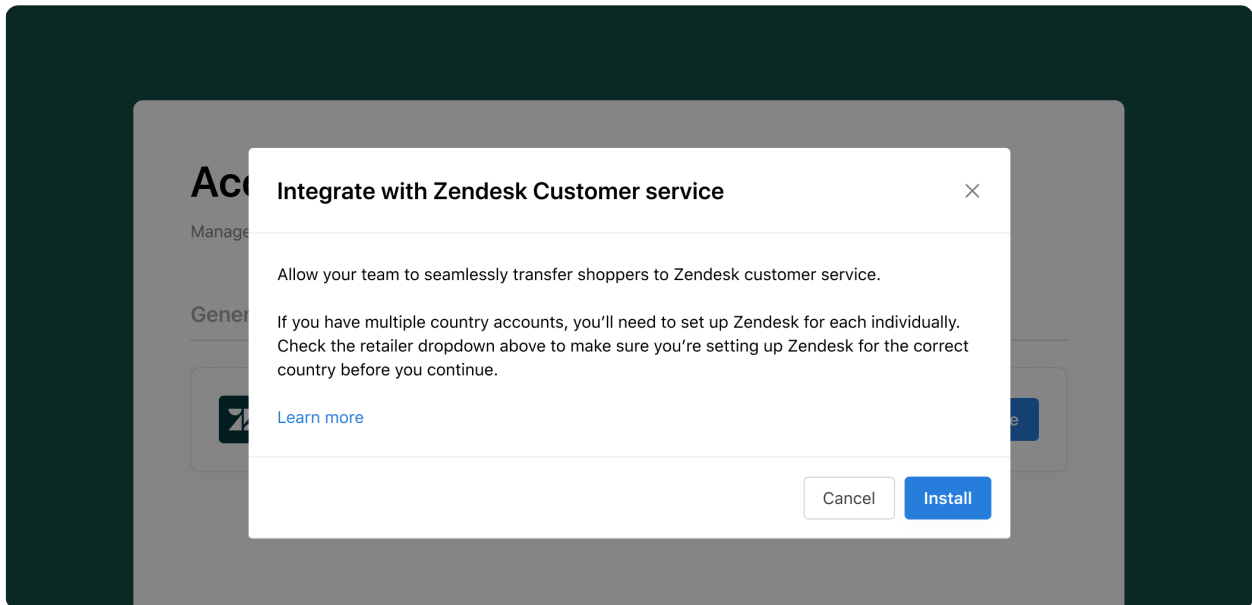
2. In the Dashboard, go to the Account Settings tab for the website and region you want to integrate with Zendesk. Choose 'Integrations' at the top of the page



3. Click 'Learn More' next to the module for the Zendesk Integration




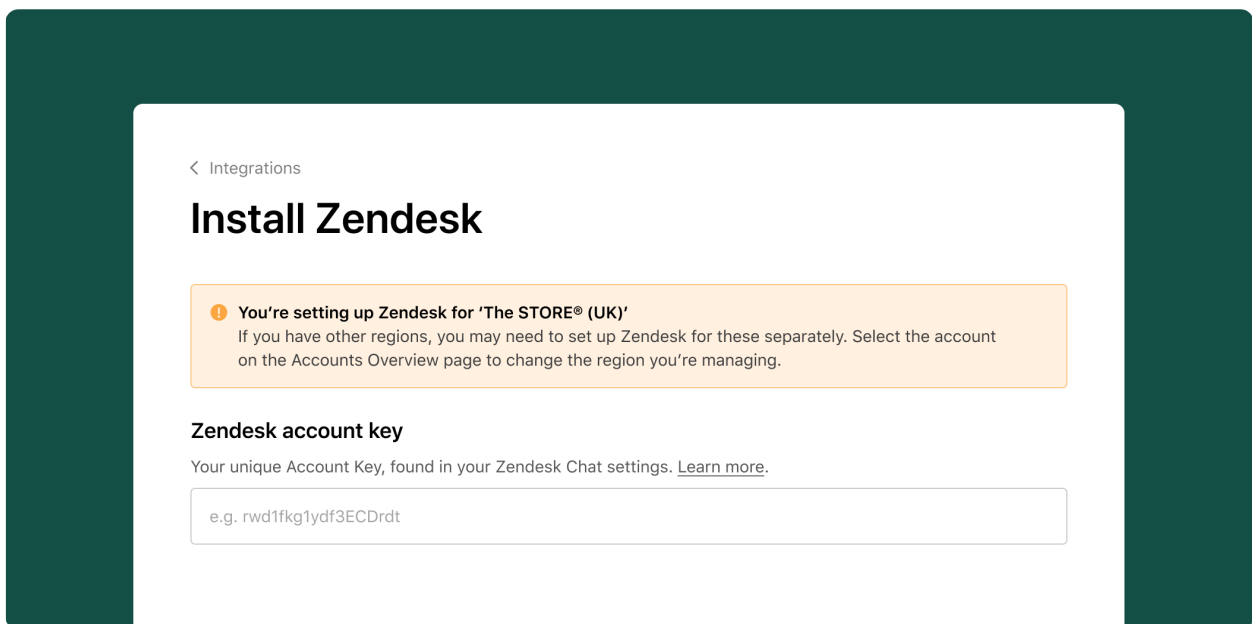
4. Now click on 'Install' to begin the activation flow



Step 2: Activating your integration

1. Enter your Zendesk Account Key to connect your Zendesk account to Virtual Shopping, click 'Next'

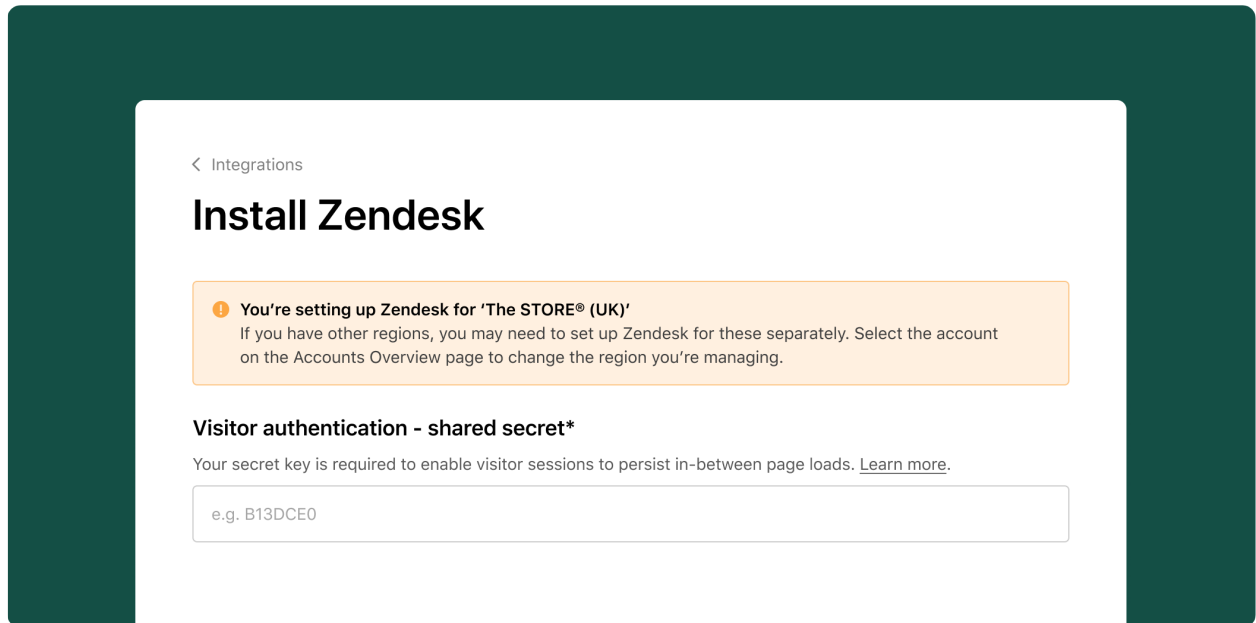
 To find your **Zendesk Account key**, visit Zendesk documentation [here](#).



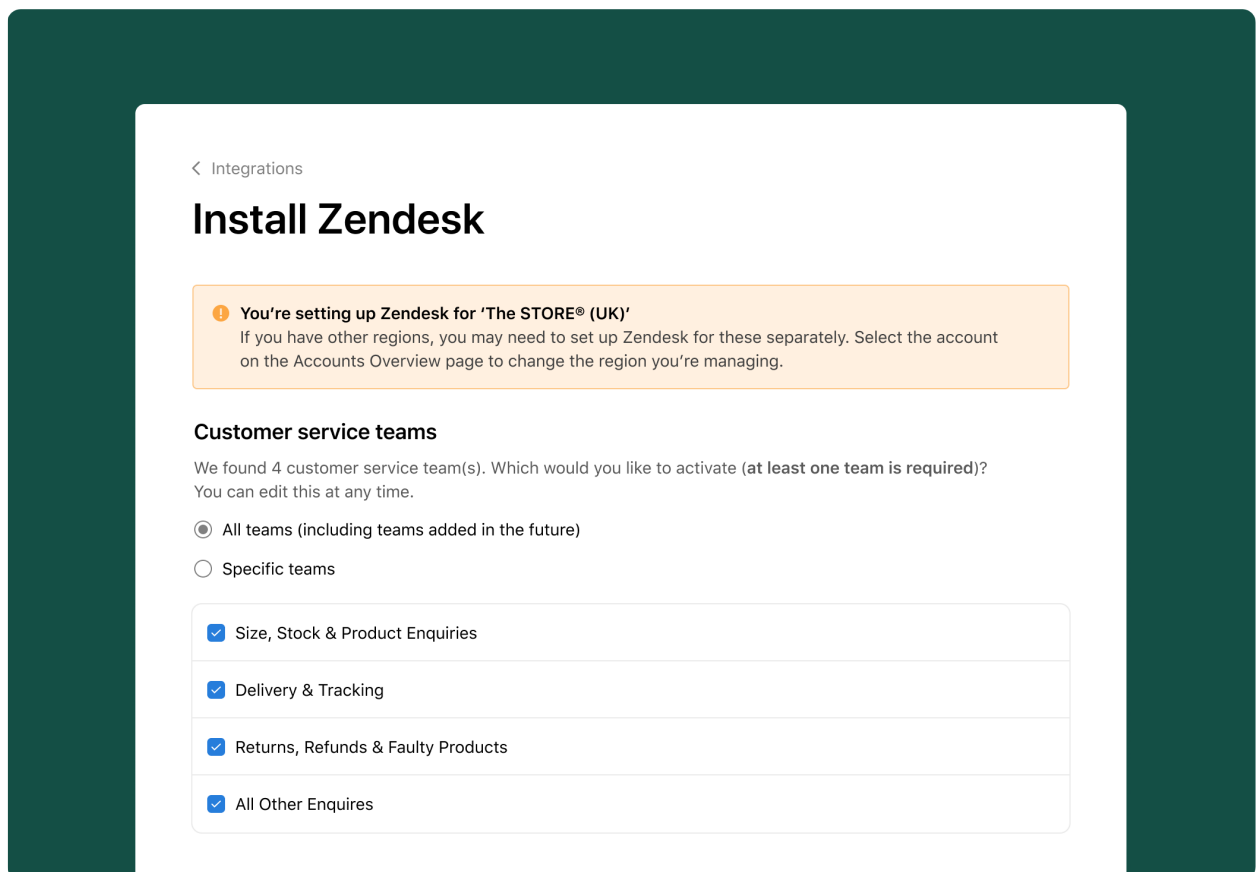
2. Enter your chat shared secret in order to persist Shopper conversations, click 'Next'

 To find out how to **generate a chat shared secret**, visit Zendesk documentation [here](#).

Please note: if at a later point you ever update your shared secret, you will need to uninstall the integration and then reinstall it using the new shared secret.



3. Here you can see which of your Customer Service teams are configured in your Zendesk account. Choose which Customer Service teams you'd like to display in Virtual Shopping for shoppers to choose from. When you've selected the Customer Service teams, click 'Confirm'



4. You will now be returned to the 'Integrations' page from your Account Settings and will see the integration status as 'Active'. **Nice job! Your Zendesk integration is now live**

Account settings

Manage your organisation's Virtual Shopping installation

General Departments Integrations



Zendesk Customer Service

Allow team members to transfer chats to agents using Zendesk

• Active



How to install the Zendesk integration.mp4 15MB

Binary

Managing your integration

Account settings

Manage your organisation's Virtual Shopping installation

General Departments Integrations



Zendesk Customer Service

Allow team members to transfer chats to agents using Zendesk

• Active



Manage

Uninstall

1. To manage your current Zendesk integration settings, click on the icon with 3 dots from the 'Integrations' page of your Account Settings and select 'Manage'

2. Here, you can choose to edit your Zendesk Account key, chat shared secret, or linked Customer Service teams. You can also choose to 'Deactivate Zendesk' if you want to deactivate the integration

3. Once you've made any updates, click 'Save changes'



Managing the Zendesk Integration.mp4 11MB

Binary



Deactivate or reactivate the Zendesk Integration.mp4 12MB

Binary

Uninstalling your integration

Once you uninstall your Zendesk integration, you will no longer be able to transfer shoppers to Customer Service teams using Zendesk.



If you plan to uninstall your Zendesk integration, please inform your Partnership Success Manager and request that the Customer Service Need is either switched off or changed to a redirect to a Customer Service page on your website by raising a ticket to help.virtualshopping@klarna.com.

1. To uninstall the Zendesk Integration, click on the icon with 3 dots from the 'Integrations' page of your Account Settings and choose 'Uninstall.' Then, in the window that appears, confirm you want to uninstall the Zendesk integration
2. From the 'Integrations' page, check your status is no longer set as 'Active'



Uninstall the Zendesk Integration.mov 3MB

Binary

Update customer details

The Virtual Shopping Client API exposes an `update` method which can be used to provide Virtual Shopping with relevant data such as a customer email address or a customer ID number. This would enable you to tie Virtual Shopping data back against your own by ensuring there is a consistent data key.

⚠ There are a number of templated Shopper data exports which can be set up by our Support team and sent to an SFTP. To have this arranged please raise a support ticket through your CSM.

Events

Update event fields:

Name	Value	Type	Required	Description
<code>customer</code>	See customer subsection	Object	yes	Customer details

Customer object subsection fields:

Name	Value	Type	Required	Description
<code>id</code>		string	no	Customer ID
<code>firstName</code>		string	no	Customer first name
<code>lastName</code>		string	no	Customer last name
<code>email</code>		string	no	Customer email address
<code>phoneNumber</code>		string	no	Customer phone number (E.164)
<code>metadata</code>		object	no	Additional customer data

Update event example:


```
hero("update", {  
  customer: {  
    id: "123",  
    firstName: "First",  
    lastName: "Last",  
    email: "example@example.com",  
    phoneNumber: "+447901234567",  
    metadata: {}  
  }  
});
```

Set up Contacts

Growing customer relationships with Contacts

Using Virtual Shopping Contacts, your team is able to build a long-term relationships with customers. Once a customer has shared the contact details for their preferred channel teams can keep the conversation going over text, email, or WhatsApp to encourage repeat purchases, all via the Store app.

To set up email or SMS for Contacts, please get in touch with your Customer Success Manager, and Klarna can switch on these channels and support your team with training.

To set up WhatsApp for Contacts there are some dependencies on you; first we'll review the shopper journey and then the steps to integrate:



How WhatsApp for Contacts works



Set up WhatsApp for Contacts



Please note: WhatsApp for Contacts should only be used for one-to-one chats. WhatsApp does not currently approve marketing campaigns, so please ensure you do not use this service for mass marketing to avoid the risk of the service being blocked.

How WhatsApp for Contacts works

What happens in a chat over WhatsApp

1. First, a team member invites a shopper to give their contact details so they can begin chatting over WhatsApp



10:00 AM



the-store.co



Tessa
Women's



Of course - I'll send you an invite to become a contact and let you know when they're back in stock



Stay in touch with Tessa

Connect with Tessa for updates and advice, whenever you need it.

Email

SMS

WhatsApp

 +44|12345678



I agree to receive messages from The Store. See our [Privacy Policy](#) for more details.

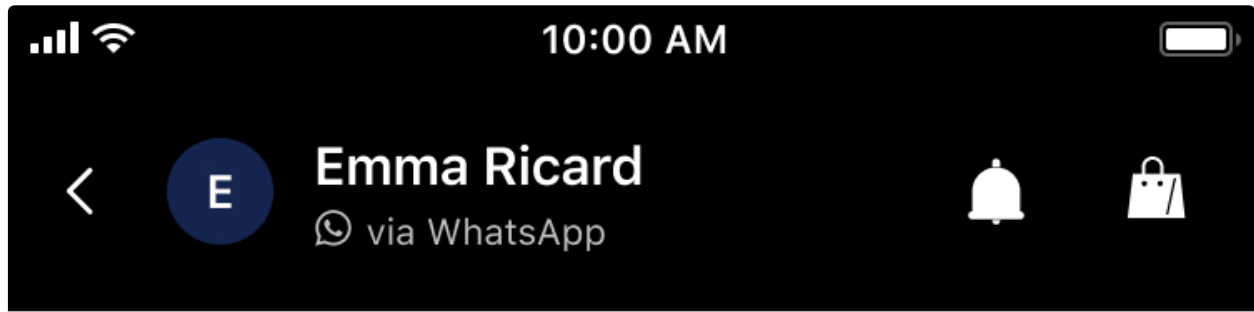


Type a message...





2. The shopper then receives a WhatsApp message asking them to opt-in



Hi Emma, this is Tessa from The Store. Thank you for connecting with me. Click this link to finish opting in <http://s-f.me/1234>

Auto-sent by Hero

Waiting for opt-in so you can send messages to Emma

3. Following a successful opt-in by the shopper the team member can contact the shopper at any time by choosing a templated message. Virtual Shopping will work with you to create suitable template messages that will be approved by WhatsApp.



10:00 AM



Select message

Send a template message to get a response from Emma

Follow up



Hi Emma, It was great to speak with you. Let me know if I can assist with anything.

Back in stock



Hi Emma. Just to let you know, an item you were looking for is back in stock.

Item Recommendation



Hi Emma. I have selected an item for you, as discussed. Let me know if you would like me to share it with you.

Event in-store



Hi Emma. As discussed, there is an event happening in-store. Would you like to



Send

4. Once the shopper replies, the team member and shopper can chat over WhatsApp freely for 24 hours



10:00 AM



Emma Ricard

via WhatsApp



Hi Emma, this is Tessa from The Store. Thank you for connecting with me. Click this link to finish opting in <http://s-f.me/1234>

Auto-sent by Hero



Emma has opted in

Hi Emma. I have selected an item for you, as discussed. Let me know if you would like me to share it with you.



That's great, yes please!



Message...



To set up WhatsApp as a channel for Contacts follow the guidance below:



Set up WhatsApp for Contacts

Set up WhatsApp for Contacts

Setting up WhatsApp for Contacts

1. Prepare your application to WhatsApp


Before getting started, Virtual Shopping will need send an application to Facebook to activate WhatsApp as a channel for messaging customers via the Store App.

So that Virtual Shopping can submit this application, please take the following steps:

1. Send Virtual Shopping your Facebook Business ID. Find out how to source this ID [here](#)
2. [Create a Twilio Console account](#) and accept the Terms of Service. In the Twilio Console, navigate to 'Programmable SMS', select 'WhatsApp' and choose 'Accept' from the dropdown
3. Send Virtual Shopping the Account Security Identifier (SID) for your Twilio Console account

Virtual Shopping will then:

- Set up a sub-account for you in Virtual Shopping's Twilio Console
- Configure the supplied phone numbers for WhatsApp usage in Twilio Console
- Once the form is complete, submit to Facebook for approval

 Please note: your WhatsApp account can support up to 25 numbers. Virtual Shopping will need to use only one of these as routing is handled seamlessly in the background

2. Approve the request from Twilio to send messages on your behalf

Virtual Shopping will let you know once the application has been submitted to WhatsApp. Once it has been approved, please take the following steps:

1. Go to business.facebook.com and click 'Business Settings' to the left of your screen under 'Home'. To the left of your screen, choose 'Requests' under the key icon. Select 'Invitations' and choose 'Accept'. WhatsApp will review your Display Name within 3 days, as soon as your Facebook Business Manager account is verified and you have accepted Facebook's invitation to allow Twilio to send messages on your behalf
2. Once WhatsApp has approved your Display Name and Virtual Shopping has configured your phone numbers, Virtual Shopping will submit 'Templates' for approval
3. Once the templates are approved, your team can begin sending customers messages using these Templates
4. If you would like to change Templates in the future, Virtual Shopping can submit others for review at any time




Verified Account status


It's good practice to try to set up a Verified Account so that when customers receive a WhatsApp message from your team, your brand's profile icon will appear with a green checkmark and show your brand name instead of a phone number.

Virtual Shopping can ask Twilio to submit a request to have your WhatsApp profile considered to be listed as 'Verified' by Facebook. But please note that Twilio cannot guarantee success; at time of writing only about 10% of all WhatsApp profiles are approved as Verified accounts.

Data API

 **If you would like access to the Data API, contact your account manager and they can help get you started.**

With the Data API, you can securely consume data held by Virtual Shopping into your other systems. You could use this to grow or augment your CRM, to better understand product expert performance and calculate commission, or as data for targeted marketing campaigns.

 As the data controller, it is your responsibility to ensure that you have suitable permission to use the data for any purposes you chose to, as the data processor Klarna is not responsible for advising you on this.

Data Points Available through the Data API

Currently the Data API makes the following data points available:

Shopper (chat)

Name	Data Point	Description
ID	_id	The Shopper ID automatically issued by Klarna Virtual Shopping, can be used to identify the specific shopper across data points
Location	location	The location of the user, stored as a GPS co-ordinate
Created date	_created_at	The date the user was created in Klarna Virtual Shopping's system. This will be when they started their first chat.
First Name	firstName	The first name provided by the user when they started their chat, this is optional so may not be stored
Shopper Name	shopperName	The default name attributed to an anonymous shopper
Last online time	lastOnlineTime	The last time that Virtual Shopping logged activity for the user on your website
User agent	userAgent	The browser, device and operating system that the user first used Virtual Shopping from
Username	username	Another form of ID that is used to link specific messages to a shopper
Visitor	visitor	Another form of ID that is created for a user before they have been issued a Shopper ID

Shopper (contact)

Name	Data Point	Description
ID	_id	The Contact ID automatically issued by Virtual Shopping, can be used to identify the specific shopper across data points.
User ID	userId	The Shopper ID automatically issued by Virtual Shopping, can be used to identify the specific shopper across data points.
Acquisition date	acquisitionDate	The date the shopper was added to Contacts
Updated date	_updated_at	The date that the user record was last updated or a message was sent or received
Associate ID	associateId	The Associate ID of the Product Expert the Shopper is currently assigned to
First Name	firstName	The first name provided by the user when they were added to Contacts
Last Name	lastName	The last name provided by the user when they were added to Contacts
Phone Number	phoneNumber	The user's telephone number depending on the channel they have been added by this may not be present
Email	email	The user's telephone number depending on the channel they have been added by this may not be present
Birthday	birthday	The date of birth of the user. Optional

State	state	Covers the full variety of state the user could be in, including opted state or re-assigned
Opt-in state	optInState	The opted state of the user, indicating if they have finished opting in or have opted out
Notes	notes	Any notes captured by the product expert about the shopper

Product Expert

Name	Data Point	Description
ID	_id	The Associate ID of the product expert
Approval Status	approvalStatus	The approval status of the user indicating if they are approve claim chats or not
Email	email	The user's registered email address
First name	firstName	The user's first name as given during registration
Last Name	lastName	The user's last name as given during registration
Online status	available	The status indicating if they are online in the app
Chat status	busy	The status indicating if they are in a chat
Created at	_created_at	The date the user started registration
Spoken Languages	spokenLanguages	The languages spoken by the user as selected when creating or editing their profile
Store name	storeName	The user's currently assigned store
Store ID	storeId	The ID in Virtual Shopping's systems of the user's currently assigned store
Employee ID	employeeId	The employee ID of the user provided by them during registration

Shopper events

You can retrieve Shopper events tracked by the frontend implementation (Related section: <https://docs.virtual-shopping.klarna.com/implementation/track-shopper-events>)

Name	Data Point	Description
ID	_id	Event ID
Type	type	Event type
Time	time	Date and time when the event occurred
User ID	userId	The Shopper ID automatically issued by Virtual Shopping, can be used to identify the specific shopper across data points
Session ID	sessionId	Analytics session identifier which can be used to group events into a browsing session
Product SKU	productSku	Product SKU provided by the frontend tracking <i>(available for: user-product-view, user-basket-add, user-basket-delete)</i>
Product Quantity	productQuantity	Quantity of product added to basket <i>(available for: user-basket-add, user-basket-delete). Will be available depending on frontend implementation.</i>
Category	category	Name of category that was viewed by the shopper <i>(available for: user-category-view). Will be available depending on frontend implementation.</i>
Search terms	searchTerms	Search terms entered by the shopper <i>(available for: user-search). Will be available depending on frontend implementation.</i>
Page location	location	URL of the page of the event <i>(available for: user-category-view, user-search). Will be available depending on frontend implementation.</i>
Attributed Associate ID	attributedAssociate	ID of associate who is attributed the sale - will not be set if the order is not attributed to anyone

Order ID	orderId	Unique order ID provided by frontend implementation. It should match the order ID in your system
Order subtotal	subtotal	Order subtotal
Order currency	currency	Order currency

Steps to implement the Data API

1. Review the data points available and consider how you want to use the data and in what systems
2. Use the [technical documentation](#) to understand how to authenticate with the Data API and how to integrate it. Get estimates from the developers who will be doing the work.
3. Contact help.virtualshopping@klarna.com who can activate permission for you to implement the API and provide you with your client ID, client secret, and the endpoint to use.
4. Use the [technical documentation](#) to implement the API

Planned updates to the Data API

We plan to continuously work to advance and better our Virtual Shopping Data API.

Here is an idea of what's to come:

Phase 4 - Conversation text message data available as well as all other conversation message data, including recommended products and media

Phase 5 - All performance data available

FAQ

Why would I use this?

Our API gives you detailed insights that can be used for performance tracking as well as targeted ad campaigns.

How hard will this be for my team to implement?

We have a user-friendly set of API docs to ensure the integration process is as simply laid out as possible. Don't forget, we are here to help too (help.virtualshopping@klarna.com). Time to implement varies by merchant.

How do I get access to this?

Please contact your customer success manager if you wish to gain access to the API. They will be able to arrange for you to be provided with a token to get you started.

Is there a cost associated with it?

No, there is no cost associated with this.

What do I have legal rights to use this data for?

As the data controller, it is your responsibility to ensure that you have suitable permission to use the data for any purposes you chose to, as the data processor Klarna is not responsible for advising you on this.

Test Virtual Shopping

Overview

Ensure your Virtual Shopping implementation is working correctly

Once you have added the Virtual Shopping JavaScript to your pages and set up Shopper Event tracking, we recommend you test that On-site Chat is working on your website and that your configurations are set up correctly.

To help you do this we've put together this simple guide to help you feel confident that Virtual Shopping is behaving as expected and solve common challenges that may arise during set up.

In this section you'll learn:

- ✔ **How to test Virtual Shopping tag**
- ✔ **How the Inspector Tool works**
- ✔ **How to test Shopper Event tracking**
- ✔ **How to troubleshoot common issues**

To test Virtual Shopping, you will need to start a real chat with On-site Chat on your website, and claim it on your mobile device. The Practice Chat functionality can't be used for testing purposes.

⚠ Before you can move to these tests, the Virtual Shopping team will need to set up your configurations within the launch process. Reach out to your Customer Success Manager or assigned Launch Manager before taking this step.

Get started



Inspector Tool



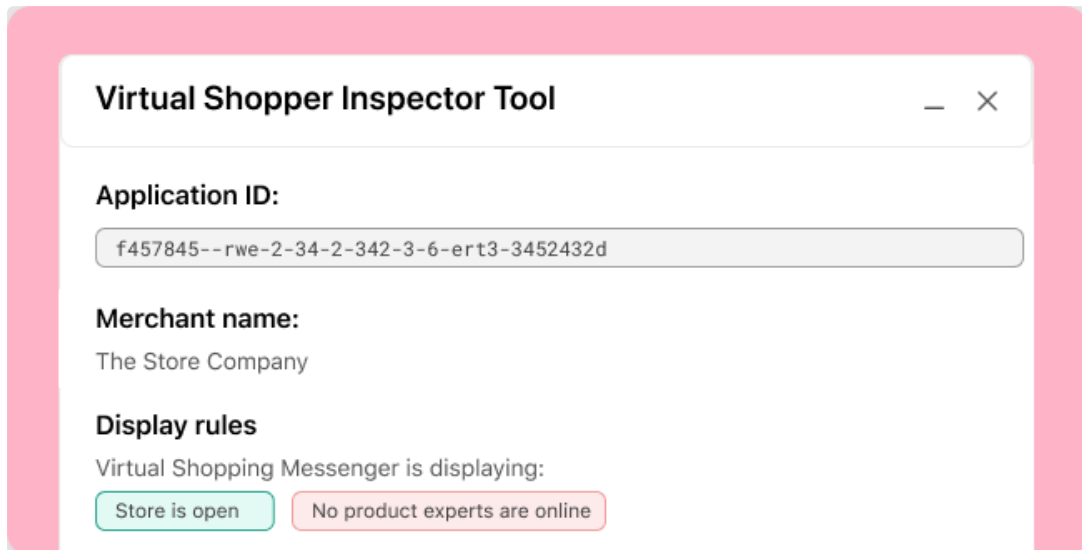
Test the Virtual Shopping tag




Test Shopper Event tracking

Inspector Tool

Quickly spot common issues while setting up Virtual Shopping 🧐



With the Inspector Tool, you can quickly spot common issues when setting up Virtual Shopping. If you are working on Virtual Shopping implementation, **we recommend using this tool to support with a smooth and efficient launch process.**

 All the information you need on how to fix issues identified with the Inspector tool can be found in the [Dev Docs](#).

The Virtual Shopping Inspector Tool loads on any website with the [Virtual Shopping Tag](#) implemented, and displays via a panel on your browser session. To learn how to use the tool and its range of capabilities, jump straight in:



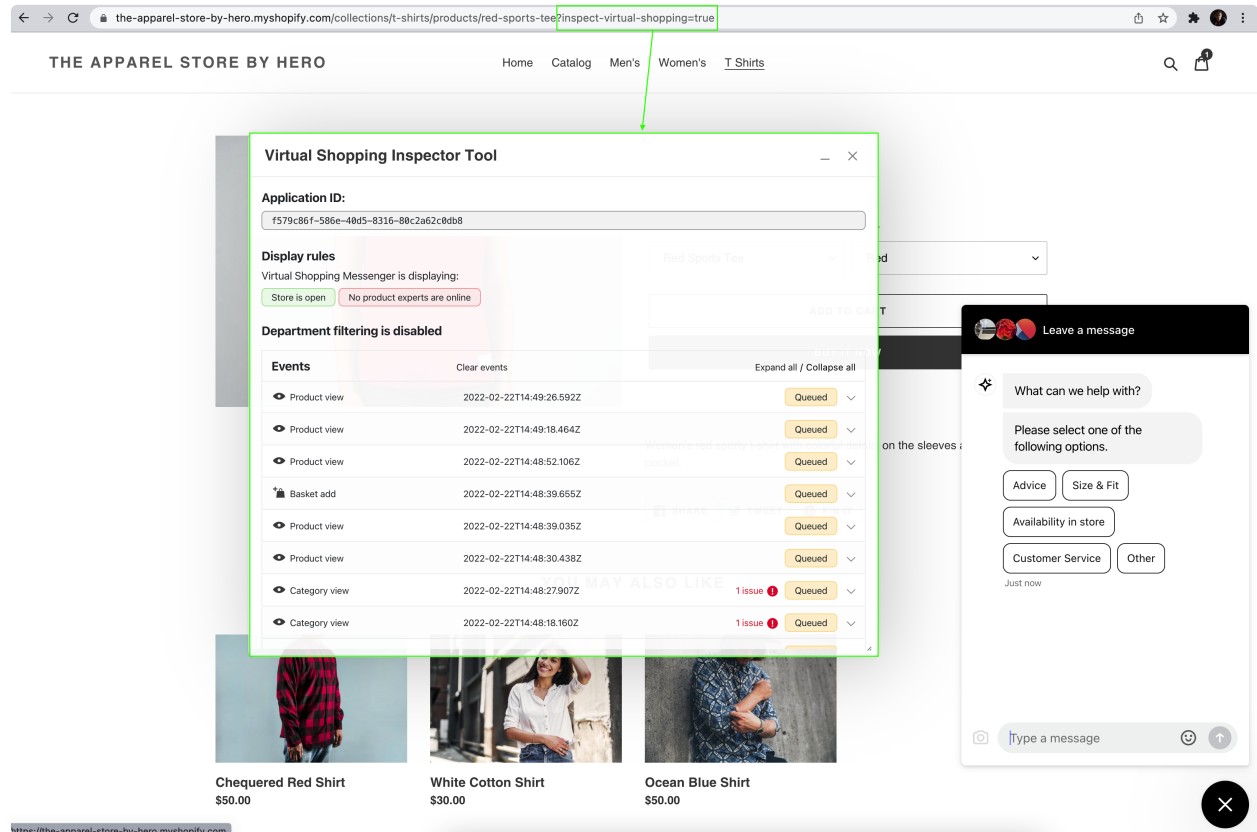
How to load the tool



What you can inspect

How to load the tool

How to open the Inspector Tool



How to load Virtual Shopping Inspector Tool

Firstly, open the website where you have added the [Virtual Shopping Tag](#) and [Shopper Events](#).

Then add `inspect-virtual-shopping=true` to the end of your URL as a query parameter, and reload the page.



For example:

<https://mywebsite.com/uk/index.html?inspect-virtual-shopping=true>

You will only need to do this once per website (domain). If you have multiple subdomains/environments open, you will need to load this separately on each browser. You will also need to do this if you clear the website cookies or each time you use Incognito Mode. After the first load of the Inspector Tool, the browser will remember that it was loaded and it will then load it automatically across all the pages of the website where the Virtual Shopping Tag is added.

What you can inspect

The Inspector Tool can check for common issues in 3 key areas:



Check Application ID & Merchant Name



Check Display Rules



Check Shopper Events

Additional features (dependent on setup) can be viewed to troubleshoot any issue:



Check Department Filtering configuration



Other Information

Check Application ID & Merchant Name

Checking if the right App ID is in place

You can use the Inspector Tool to check the **Application ID (App ID)** used to load the on-site chat on your website. This can be used to ensure the correct Staging or Production App ID has been used to launch Virtual Shopping.

The **Merchant Name**, which corresponds to the Application ID, will be displayed. This provides additional confirmation that the correct application ID has been implemented.

Application ID:

f457845--rwe-2-34-2-342-3-6-ert3-3452432d

Merchant name:

The Store Company

If no App ID has been added, the message `No Application ID found` will display. If this happens it's a good idea to check that your team has deployed the correct [App ID](#).

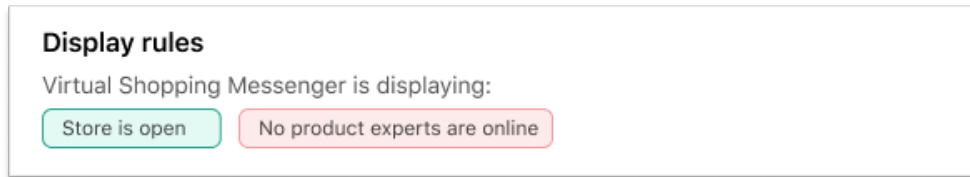


Your App IDs would have been sent to you when you began the setup process. If you are unsure what your App ID is, please get in touch with your Customer Success Manager.

Check Display Rules

Checking why Virtual Shopping is not displaying

The Display Rules section helps you understand whether or not the Virtual Shopping chat is displaying on your online store. Getting a clear view of what rules are in place you can help you quickly understand if there is an issues as a result of unwanted rules being in place, or if Virtual Shopping is behaving as expected.








Display Rules indicate:

- if the store is open/closed
- if there are product experts online
- if all product experts are busy
- if the leads queue is full
- display rules (pages allowed or denied to display the on-site chat)
- Geo-restrictions (if enabled)

Any rules that allow Virtual Shopping chat to display are shown in green. Rules that do not allow the chat to display are shown in red.

Check Shopper Events

The Events section lists all of the recorded [Shopper Events](#) since a shopper started their browsing session.

Events <small>Clear events</small>		Expand all / Collapse all	
 Product view	2023-01-10T09:51:01.244z	Queued	▼
 Purchase	2023-01-10T09:51:01.244z	Queued	▼
 Category view	2023-01-10T09:51:01.244z	3 issues 	Sent ▼
 Search	2023-01-10T09:51:01.244z	Failed	▼

Results Panel

There are 3 columns in the results panel:

Events	Timestamp	Status
The event name. If the event name is not supported, <code>Invalid</code> will display instead.	The timestamp when the event was created.	The status of the Track Shop Event.

Statuses

If issues are detected with an event, the Inspector Tool will indicate this before the status value.

There are 4 possible statuses:

Status	About
Queued	The events that have been recorded in the browsers local storage but not yet sent to Hero / This happens when the visitor is browsing the website but has not yet started a chat. If the shopper never starts a chat, these events will not be sent to Virtual Shopping. Once a chat has started the Product Expert will be able to see an other items the customer had been looking at to better understand browsing habits.
In Progress	The event is being sent to the Hero API. This is only a temporary status whilst the events are being sent to Virtual Shopping.
Sent	Events that were successfully sent to the Hero API.
Failed	Events for which the Hero API responded with an error.

Here are some examples of a **Failed** status and the reason for that failure:

The screenshot shows a list of events with the following details:

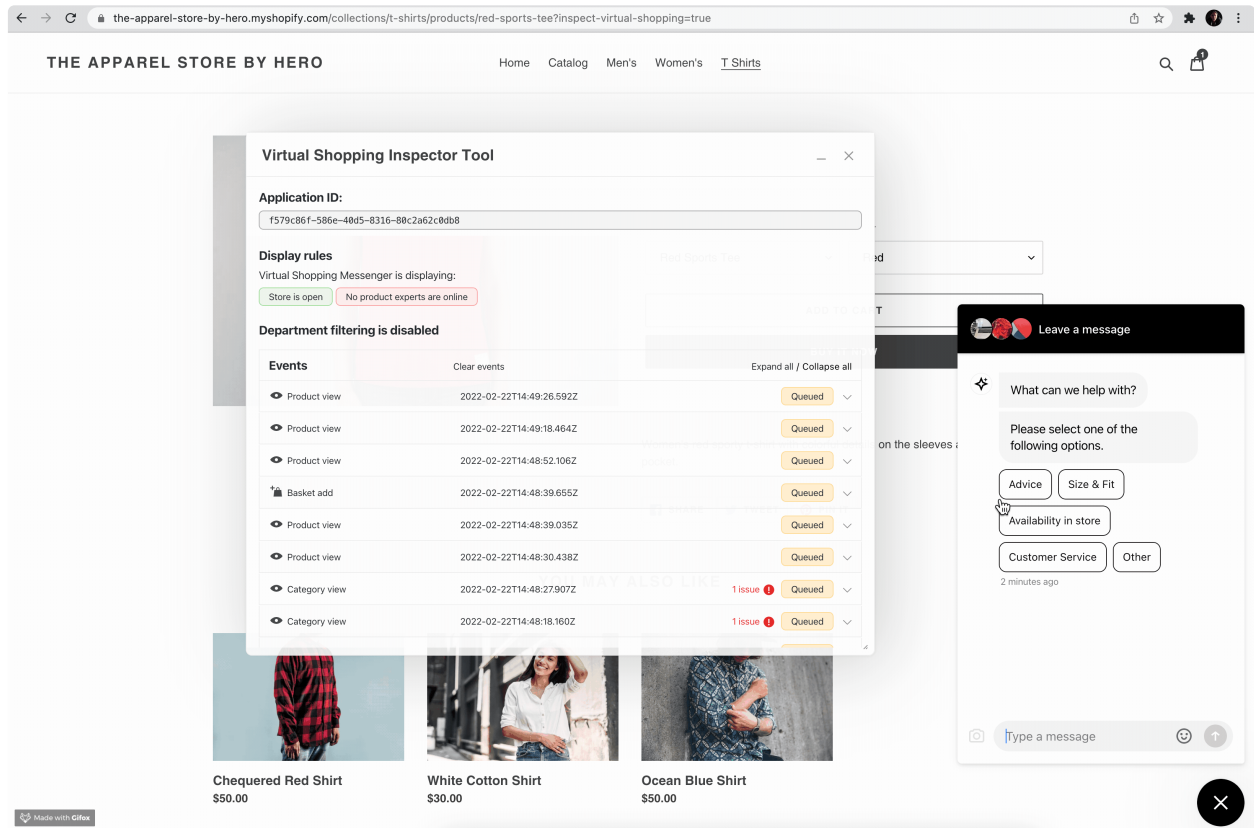
- Category view**: 2021-04-20T12:29:44.503Z, Status: Sent (1 issue icon)
- Invalid event**: 2021-04-20T11:39:35.993Z, Status: Failed (red box)
- Product view**: 2021-04-20T11:39:16.125Z, Status: Sent

The error message for the failed event is: "Invalid event type "events". Please check [our documentation](#) for the valid event types."

Failed status

Payload Details

You can expand or collapse an event by clicking on it to see more information about the payload of that event.



Subsections of a Payload

Depending on the event type, the details of an event can have multiple subsections. For example, a Product View event will have a **Product Array** subsection, and a Purchase event will have both a **Purchase Array** and **Product Array** subsection.

Each subsection has 4 columns:

Field	About
Property	The name of the field from the payload.
Type	The type of the value of the property.
Value	The value of the property. If there is no value for property a message "BLANK" will be displayed.
The validation result for the property	This can be one of the following icons: Green tick The property is valid Red X The property is invalid Dash This is an optional property for which a value was not provided

Here is an example of a valid payload without issues:

Events
Clear events
Expand all / Collapse all

Product view
2022-02-22T13:23:22.655Z
Queued

Product Details: Ocean Blue Shirt (#1606771579)

Property	Type	Value	
id	String	1606771579	✓
name	String	Ocean Blue Shirt	✓
image	String	https://cdn.shopify.com/s/files/1/0515/0402/1661/products/young-man-in-bright-fashion_925x_5f9172b9-bcd2-4ae4-99ed-098fd14843b4_925x.jpg?v=1606771579	✓
price	Number	50	✓
currency	String	USD	✓
brand	String	partners-demo	✓
category	String	Shirt	✓
variant	String	Ocean Blue Shirt / Ocean Blue	✓
location	String	https://the-apparel-store-by-hero.myshopify.com/products/ocean-blue-shirt	✓

✓ No issues found.
Copy to clipboard
Download

Here is an example of a payload with validation issues:

In this example, the `location` field has a value that is not supported as it doesn't begin with "http".

Events
Clear events
Expand all / Collapse all

Category view
2022-02-22T14:06:38.289Z
1 issue !
Queued

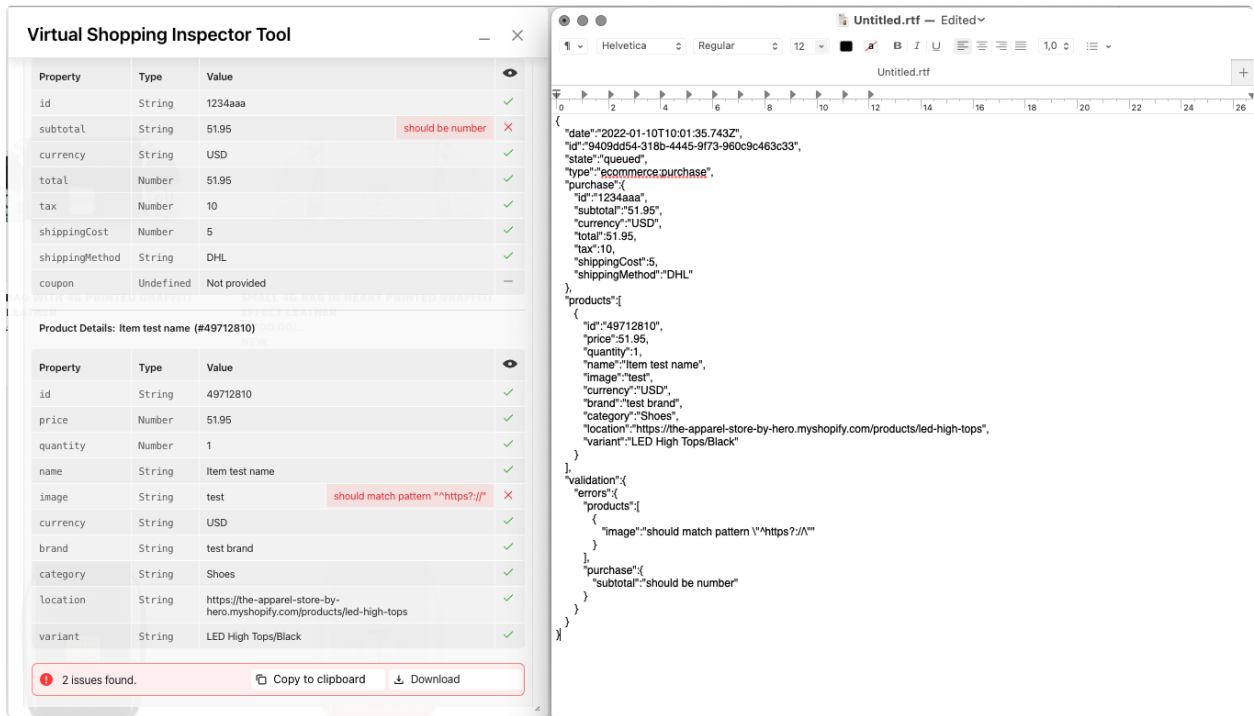
Property	Type	Value	
type	String	event	✓
action	String	category-view	✓
value	String	T-Shirts	✓
location	String	/collections/t-shirts	✗

! 1 issue found.
Copy to clipboard
Download

Copy or Download Event Payload

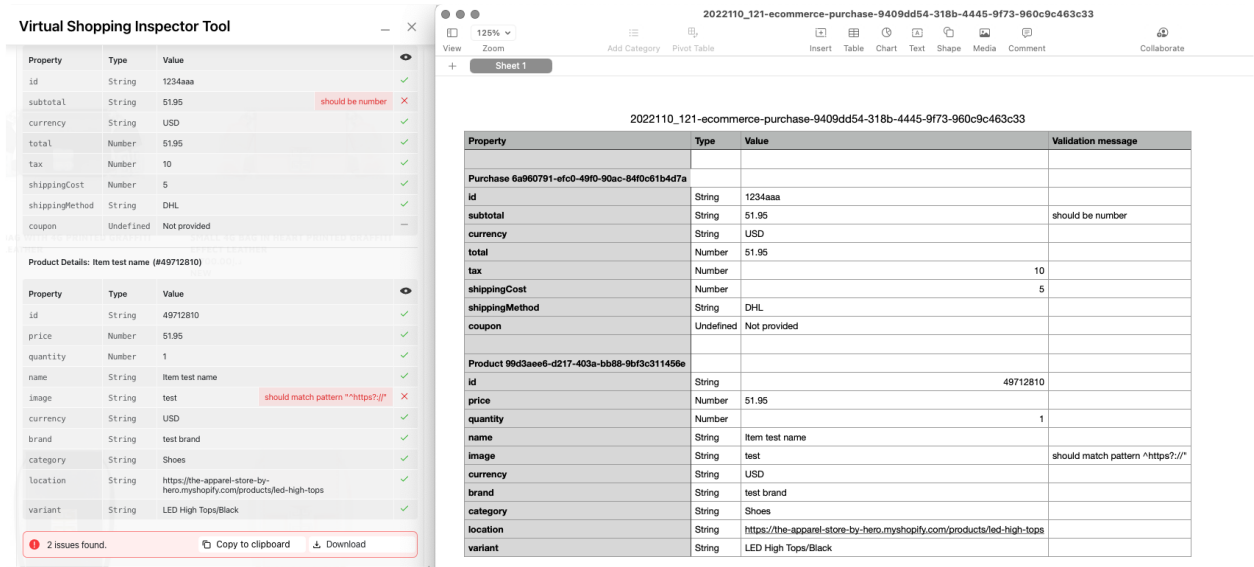
Copy Event Payload

Clicking the **Copy to clipboard** button will copy the event payload as a JSON to the clipboard such that it can be pasted into an email/ticket/document.



Download Report

Clicking the **Download** button will save a CSV file with the event details and validation messages so that the report can be easily shared via email/ticket/document.



Check Department Filtering configuration

Department Filtering can allow you to restrict which departments are shown to the shopper on the chat. The filtering can be applied to single or more pages throughout your site. More information on this configuration can be found [here](#).

If no Department Filtering is applied on the page, the Inspector Tool will indicate this.

Department filtering is disabled

Should you have Departments filtered on a particular page the Inspector Tool will display the configuration and constraints. Click on the department to see the constraint details.

Virtual Shopping Inspector Tool



Application ID:

069a5a96-e683-4c73-b6ea-d434eff736ce

Display rules

Virtual Shopping Messenger is displaying:

Store is open

Product experts are online

Department filtering is enabled

2 department(s) allowed / 0 department(s) denied

Filters

Expand all /
Collapse all

Allow

Menswear (ID: 43a0B7JDdY)

1 Constraint ▾

Allow

Women's Accessories (ID: xEK2m5oLQD)



Made with Gifox




Any errors that are detected will be described. Should you require further support with any errors please reach out to your Customer Success Manager.

Department filtering is enabled

0 department(s) allowed / 2 department(s) denied

Department with id xEK2m5oLQDX doesn't exist. Please check departments.

Filters		Expand all / Collapse all
Deny	Menswear (ID: 43a0B7JDdY)	1 Constraint 
Deny	(ID: xEK2m5oLQDX)	

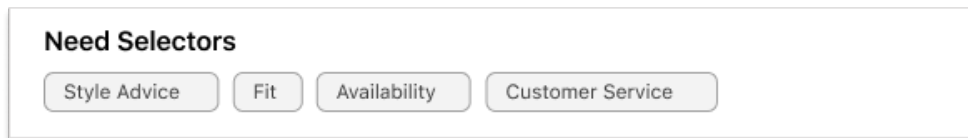
Other Information

There is additional information that Inspector Tool can show you if these are set up. This may be useful when completing the testing of Virtual Shopping, or indeed to check the setup at any time.

Need Selectors

Need Selectors are prompted to the shopper in the on-site chat to guide them through the correct chat journey. Inspector Tool will display the Need Selectors configured for your account in the order shown to the shopper.

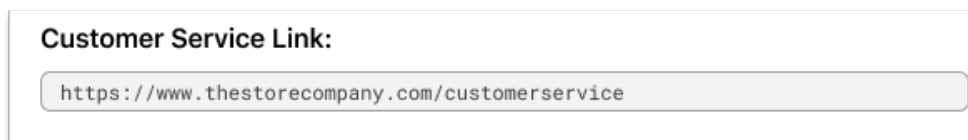
Currently, the Need Selectors cannot be managed in the Dashboard. To request any changes, please contact your Customer Success Manager.



Customer Service

Should the Customer Service link be enabled, the full URL can be seen. The link is also clickable to test if the correct URL has been set in the dashboard.

If the link does require updating this can easily be done in the dashboard, or alternatively please contact your Customer Success Manager.



Test the Virtual Shopping tag

Once you've implemented the Virtual Shopping tag, it's possible that your On-site chat will not be displaying as expected. To help you, we've put together this guidance where we'll show you:

- ✔ **How to run tests to diagnose issues**
- ✔ **Simple actions you can take to solve them**
- ✔ **Feel confident in your implementation ahead of launch**



Has the Virtual Shopping tag been deployed correctly?



Is the correct App ID in the Virtual Shopping tag?



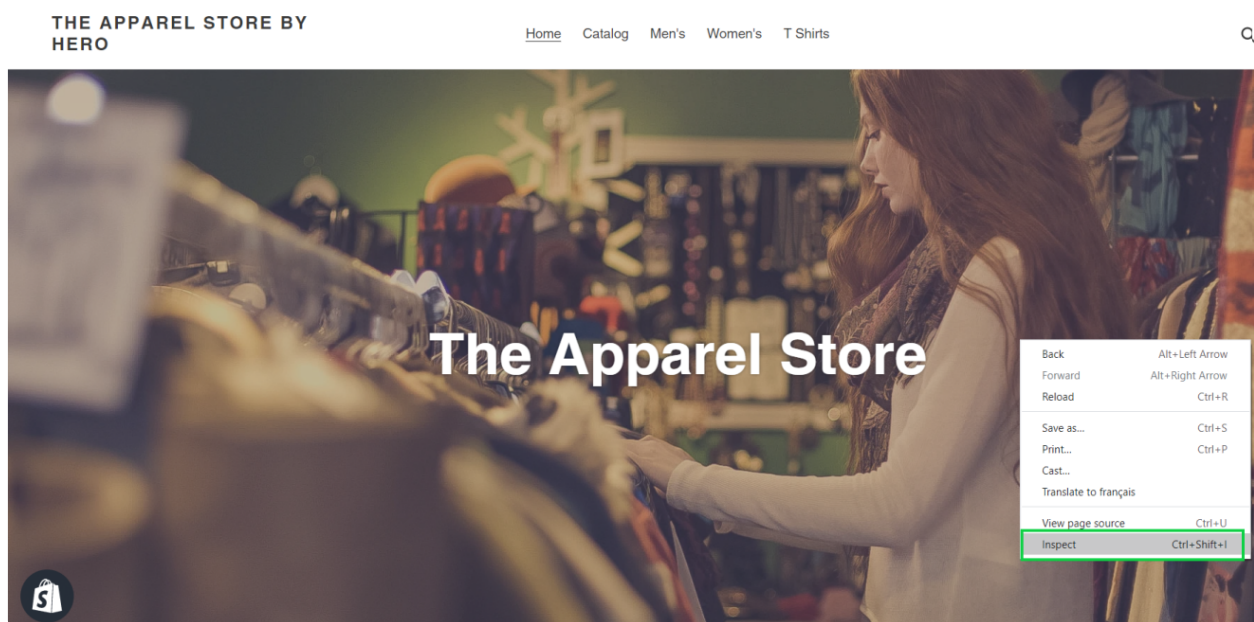
Diagnose other issues using the Display Request

Has the Virtual Shopping tag been deployed correctly?

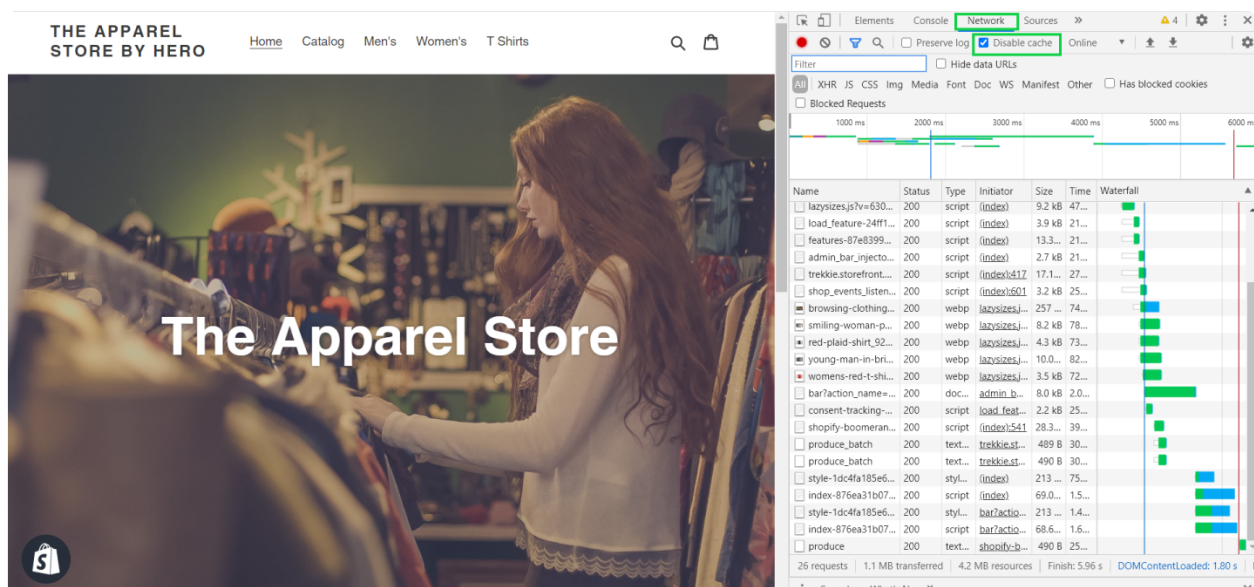
To check that you have implemented the Hero Javascript snippet correctly, simply open the Network tab in the developer tools to confirm that the Javascript is loading as expected.

Steps to test

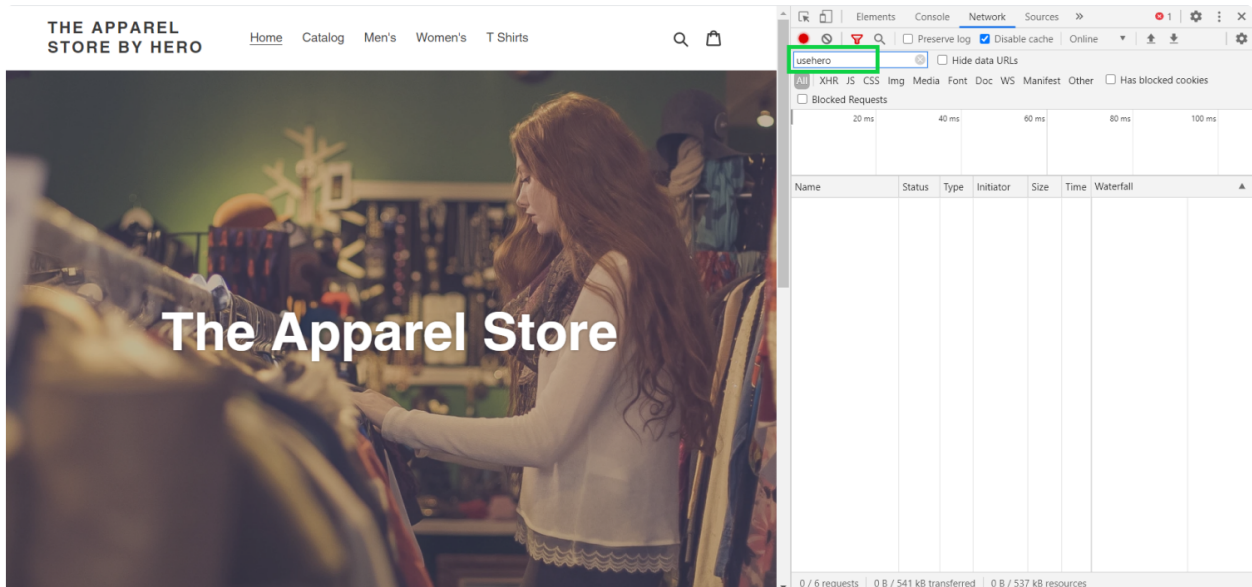
1. Click on inspect to display developer tools



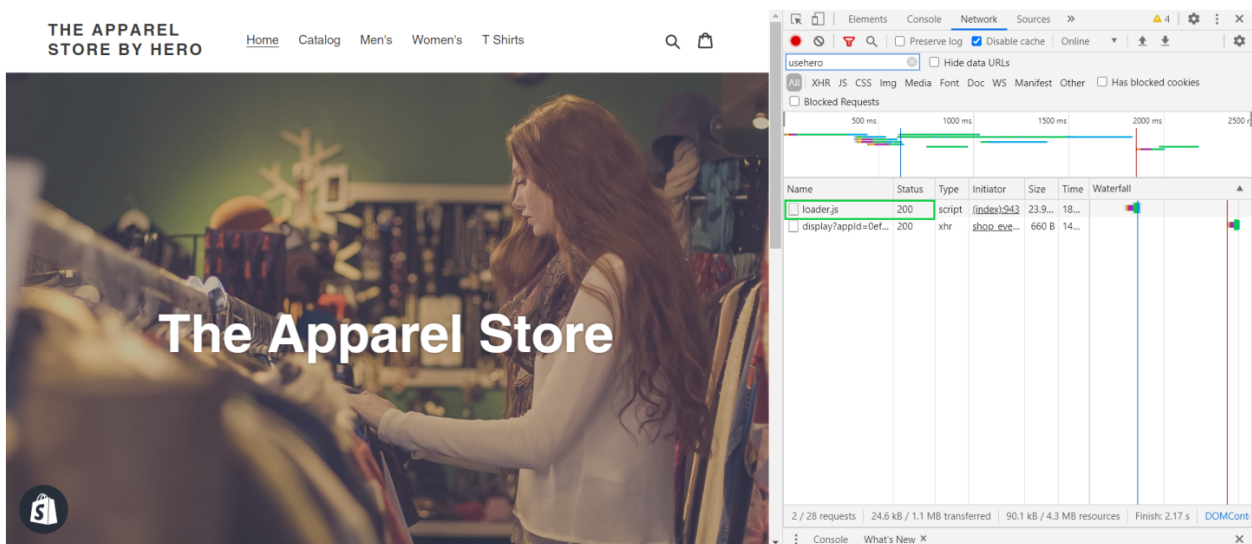
2. Open the Network tab to display Network panel. Ensure you have checked the 'Disable cache' box



3. Type in the text box: `usehero`



4. Refresh / reload the page to display network activity. Network log should display requests such as: `loader.js` and `display?appId=XXXX`



Results

Correct result:

✓ `loader.js` status column should display `200` as the HTTP response code

Incorrect result:

✗ `loader.js` request is not displayed in Network log

✗ `loader.js` does not display `200` as the HTTP response code

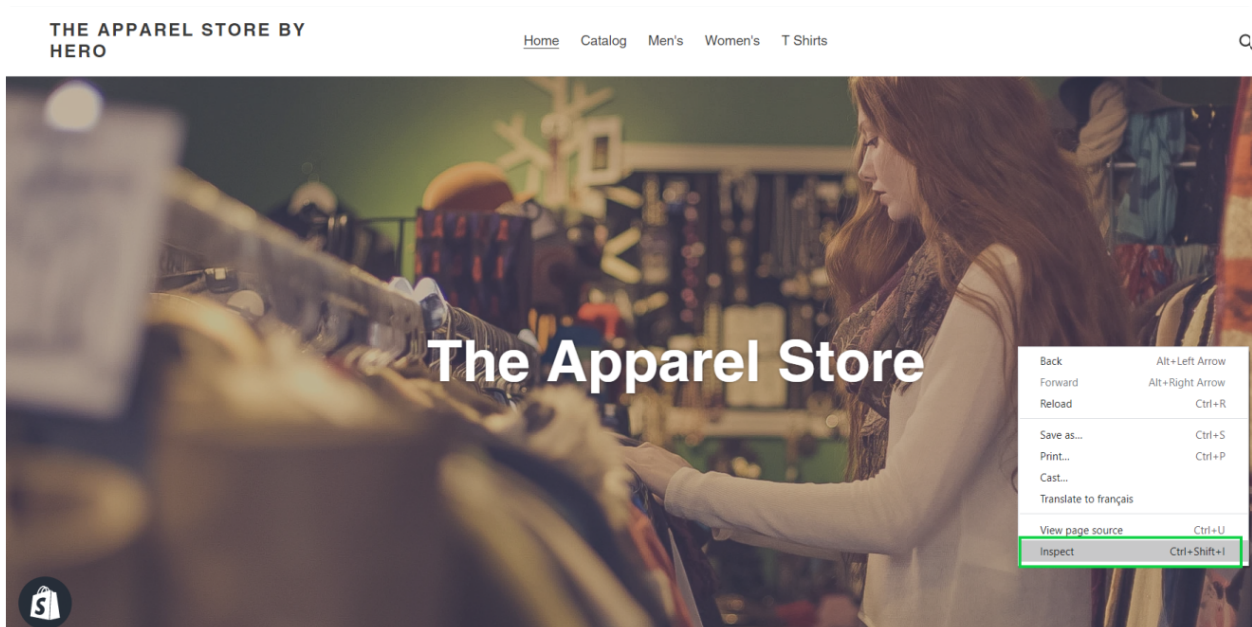
Is the correct App ID in the Virtual Shopping tag?


During set up it's possible to mix up which App ID goes where. The Staging App ID should be used for your Staging environment and Production App ID should be used for your Production environment.

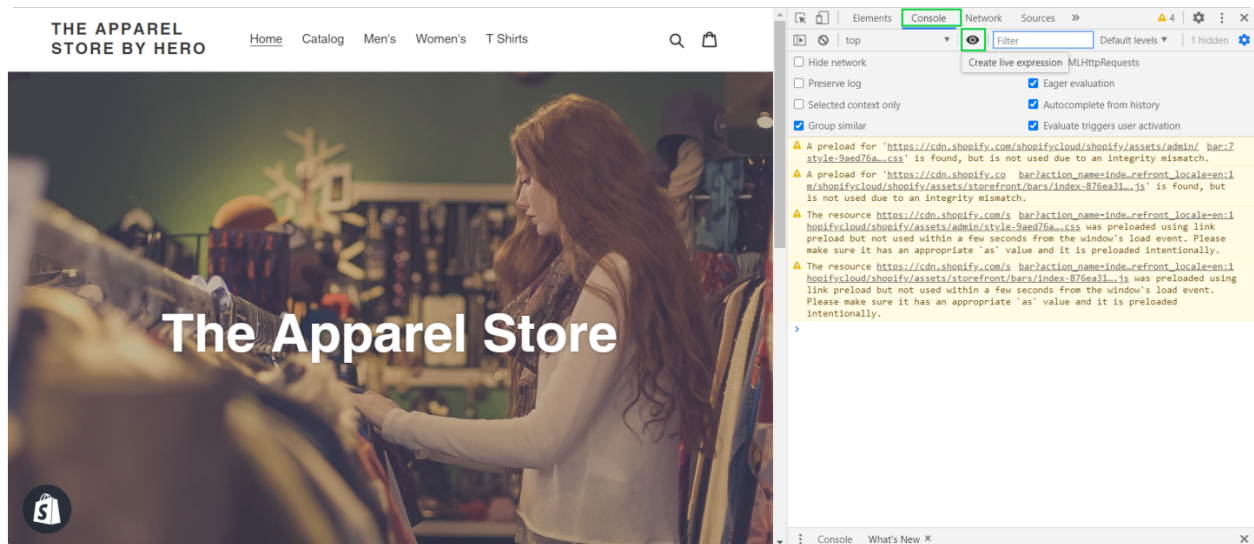
⚠️ Ensure that the correct App ID is deployed to the right environment so that Virtual Shopping knows the difference between Staging and Production. Find out more about App IDs [here](#).

Steps to test

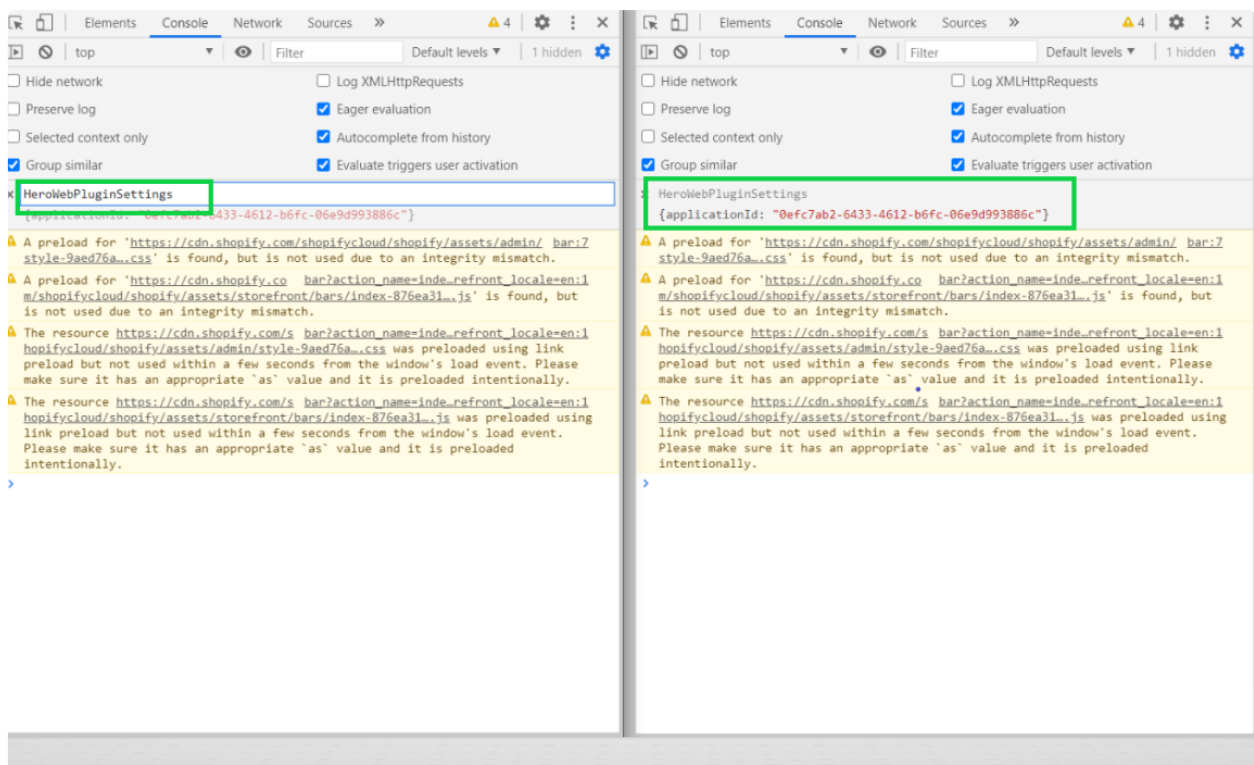
1. Right-click on website page and click on inspect to display developer tools



2. Open the Console tab 2, click on inspect to display developer tools and click 'Create Live Expression' icon 



3. Type in the text box: `HeroWebPluginSettings` then type `Control + Enter` or `Command + Enter` (Mac) or click outside of the Live Expression text box to save the expression



Results

Correct result

✔ The correct App ID implemented should be displayed.

Incorrect result

✘ The incorrect App ID will be displayed i.e Production App ID displayed on Staging environment or Staging App ID displayed on Production environment.

Diagnose other issues using the Display Request

If your Virtual Shopping On-site chat is still not appearing on your website a useful tool to help you diagnose the issue is the **Display Request** `display?appId=XXX` . The Display Request can be found in your website developer tools. Find out how to use it here:



How to use the Display Request

Get started

Now you are familiar with the Display Request, you can use it to run tests and answer these questions:



Have you deployed Virtual Shopping to all planned pages on your website?



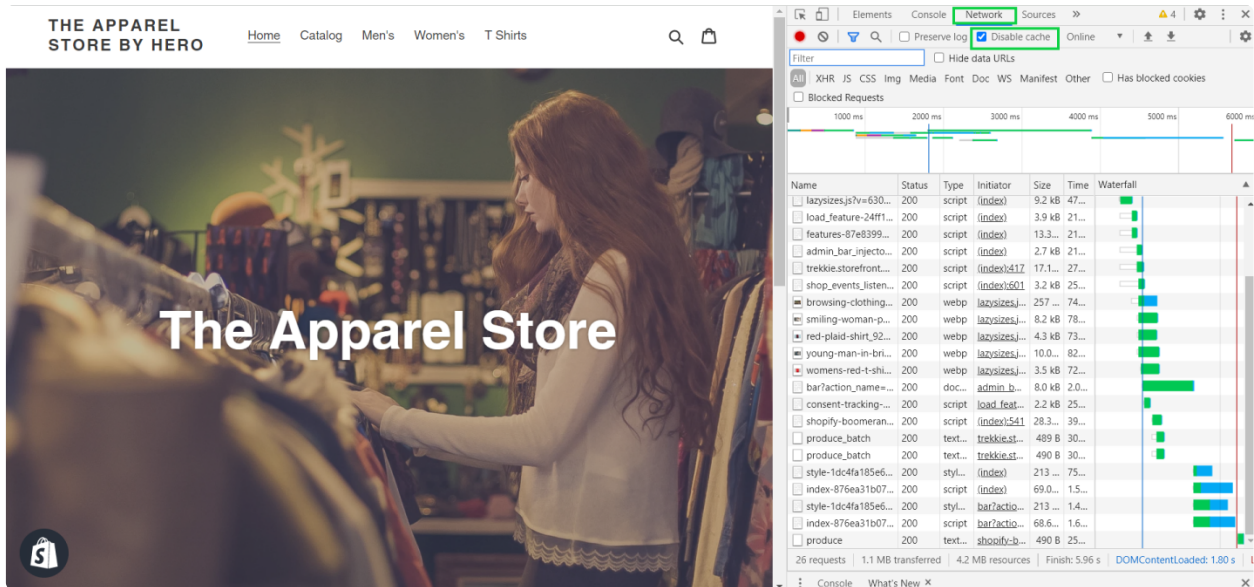
Are Dashboard configurations causing a display issue?



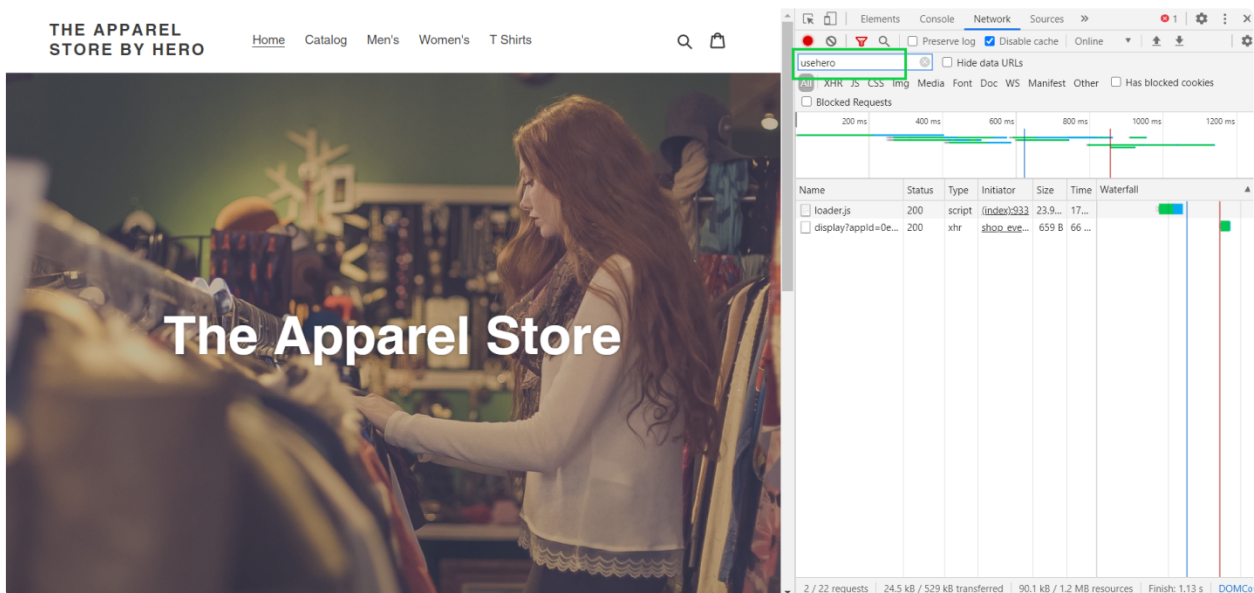
Is On-site chat being hidden by CSS?

How to use the Display Request

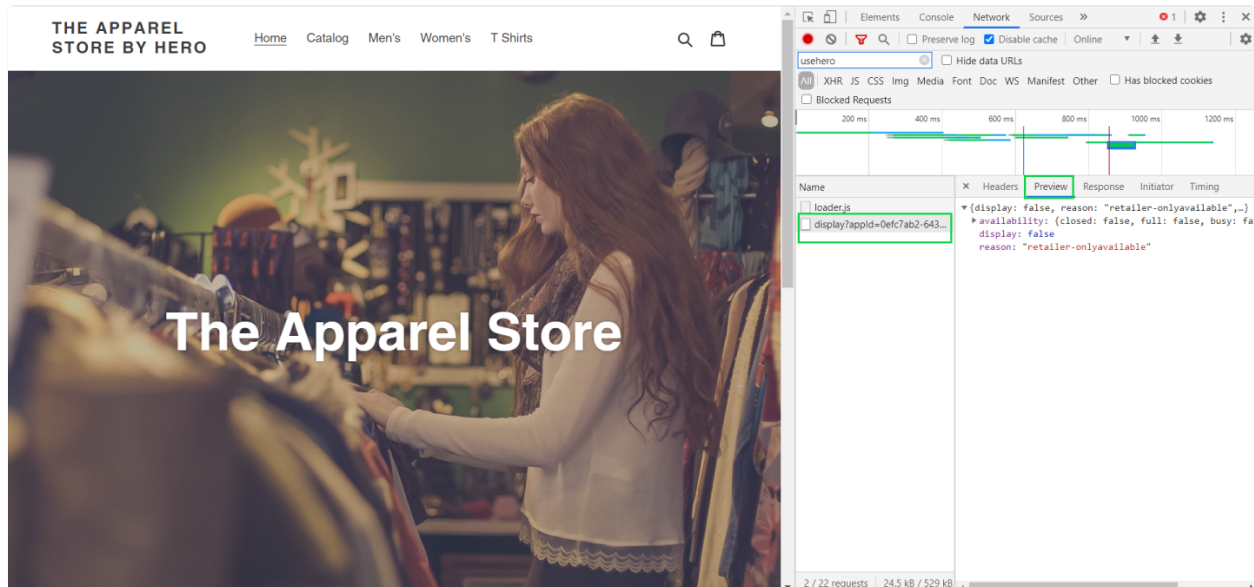
1. Right-click on website page, click on inspect to show developer tools and open the Network tab to show Network panel. Ensure you have checked the 'Disable cache' box



2. Type in the text box usehero then reload page to show network activity. The Network log should show requests such as: loader.js and display?appId=XXXX



3. Click on display?appId=XXXX then click on the Preview tab to show the Virtual Shopping Messenger display response



Correct result

✓ The Preview tab should show the current status of Virtual Shopping Messenger and why it is or is not showing

Having trouble?

If you can't get the Display Request to appear in the Network tab, follow these steps:



Why is the Display Request not appearing?

Why is the Display Request not appearing?

If the Display Request `display?appId=XXXX` is not appearing in the Network tab, this may mean that, although Virtual Shopping Messenger has been deployed to your web pages, **the App ID has not been added to the Virtual Shopping tag.**

- ✔ Ensure the App ID has been added to the Virtual Shopping tag when deploying the code across your website. Find out more about App IDs [here](#).

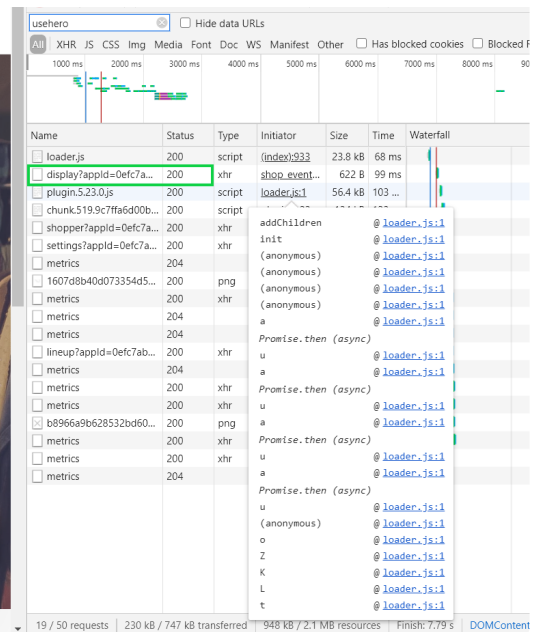
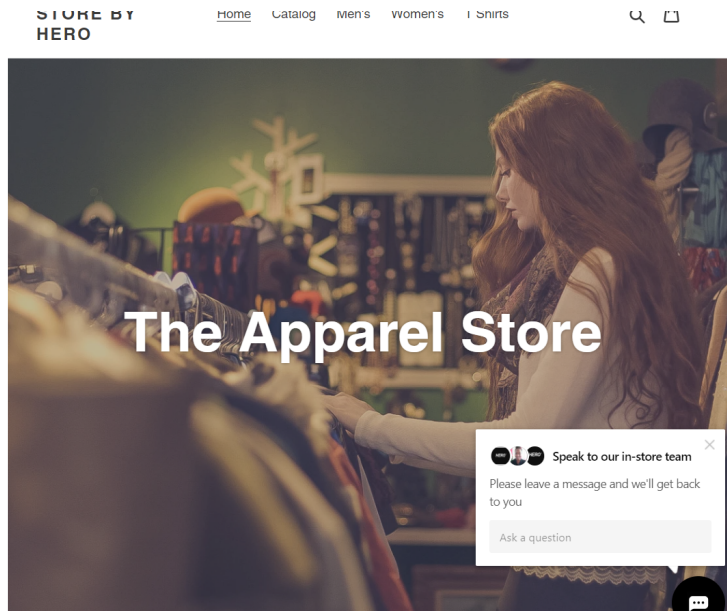
Steps to test ✔

1. Right-click on website page, click on inspect to display developer tools and open the Console tab
2. Click 'Create Live Expression' and type in the text box: `HeroWebPluginSettings`
3. Type `Control + Enter` or `Command + Enter` (Mac) or click outside of the Live Expression text box to save the expression. This will show the App ID if it has been implemented
4. Open the Network tab to display Network panel. Ensure you have checked the 'Disable cache' box
5. Type in the text box: `usehero`
6. Reload page to display network activity. Network log should display requests such as `loader.js` and `display?appId=XXXX`
7. Click on the `display?appId=XXXX`
8. Click on the Preview tab to see how Virtual Shopping has responded

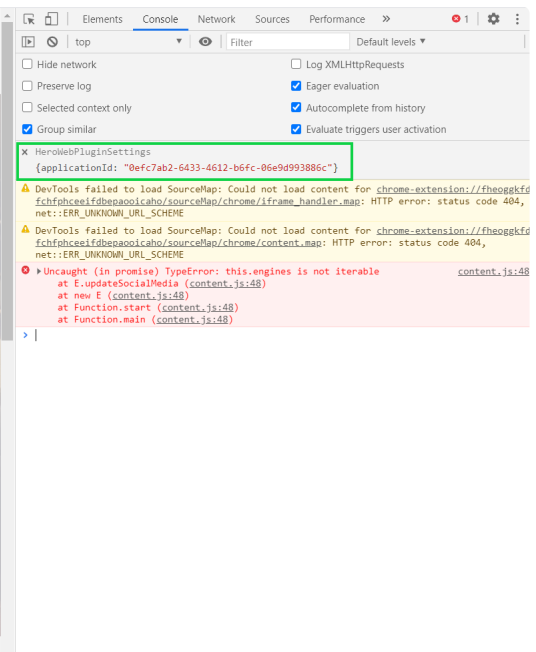
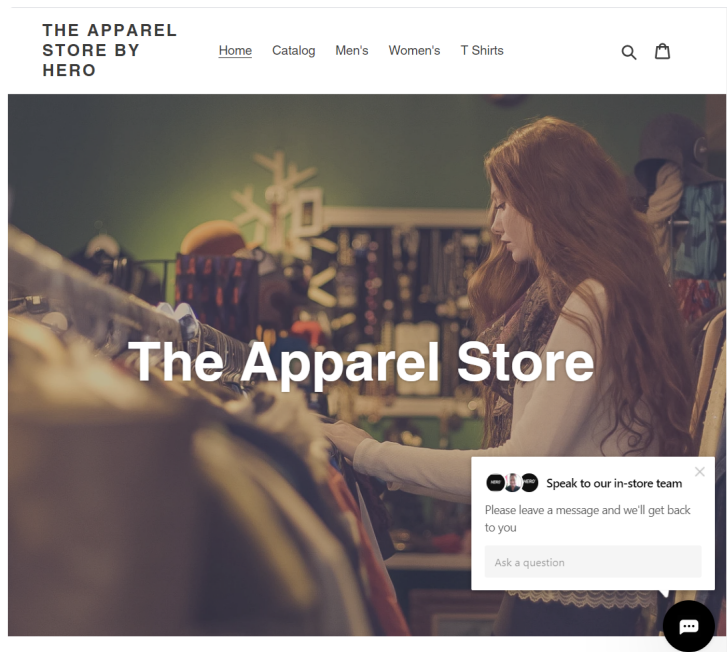
Results

Correct results

- ✔ `display?appId=XXXX` is displaying in the Network tab

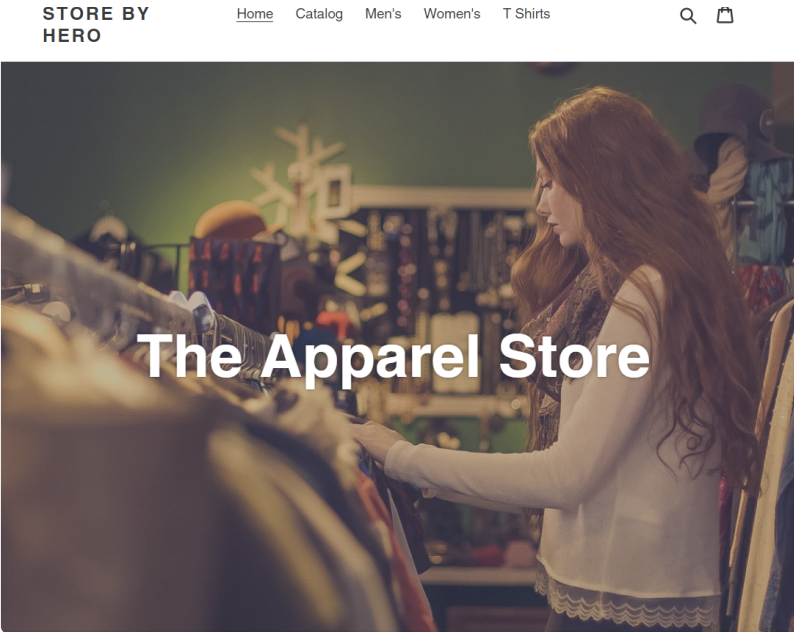


✅ App ID is displayed in the Console tab



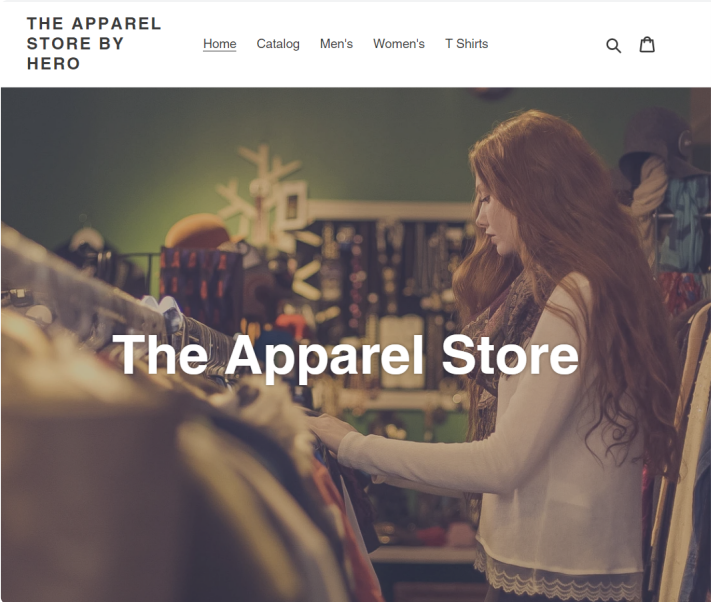
Incorrect results

❌ Display Request `display?appId=XXXX` is not displaying in the Network tab



Name	Status	Type	Initiator	Size	Time	Waterfall
loader.js	200	script	(index):945	23.8 kB	370 ...	

✗ App ID not displaying in the Console tab



Console messages:

- ✗ HeroWebPluginSettings {}
- ⚠ DevTools failed to load SourceMap: Could not load content for chrome-extension://fheogkfd.../sourceMap/chrome/iframe_handler.map: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME
- ⚠ DevTools failed to load SourceMap: Could not load content for chrome-extension://fheogkfd.../content.map: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME
- ⚠ Uncaught (in promise) TypeError: this engines is not iterable content.js:48
 - at new E (content.js:48)
 - at E.updateSocialMedia (content.js:48)
 - at Function.start (content.js:48)
 - at Function.main (content.js:48)
- ⚠ A preload for "https://cdn.shopify.com/shopifycloud/shopify/assets/admin/style/preview_bar:61e-99131e3...css" is found, but is not used due to an integrity mismatch.
- ⚠ DevTools failed to load SourceMap: Could not load content for chrome-extension://fheogkfd.../sourceMap/chrome/iframe_handler.map: HTTP error: status code 404, net::ERR_UNKNOWN_URL_SCHEME
- ⚠ A preload for "https://cdn.shopify.com/shopifycloud/shopify/assets/storefront/preview_bar:1t/bars/index-876ea31...js" is found, but is not used due to an integrity mismatch.
- ⚠ The resource https://cdn.shopify.com/shopifycloud/shopify/assets/admin/style/preview_bar:1_39151e3...css was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate "as" value and it is preloaded intentionally.
- ⚠ The resource https://cdn.shopify.com/shopifycloud/shopify/assets/storefront/preview_bar:1t/bars/index-876ea31...js was preloaded using link preload but not used within a few seconds from the window's load event. Please make sure it has an appropriate "as" value and it is preloaded intentionally.

Have you deployed Virtual Shopping to all planned pages on your website?

Sometimes, although the Virtual Shopping tag has been set up correctly, On-site chat isn't showing on all pages as expected. If you have the Network tab open and can see the JavaScript snippet is loading on one page but not on another, this may mean that **you have not yet deployed the Virtual Shopping tag to all areas of your site.**

i We recommend you add the JavaScript code snippet to every page of your website. This is so chats can follow shoppers from page to page after an initial chat is started and sales are tracked correctly.



You do not have to add the Virtual Shopping tag to sensitive pages in your checkout journey but the tag does need to be deployed to the transaction confirmation / complete page to track sales.

Steps to test

1. Right-click on website page where the On-site chat is not displaying
2. Click on inspect to display developer tools
3. Open the Network tab to display Network panel
4. Type in the text box `usehero`
5. Reload page to display network activity. Network log should display requests such as `loader.js` and `display?appId=XXXX`

Results

Correct result

-  `loader.js` is displaying in the Network log
-  Status column should display `200` as the HTTP response code

THE APPAREL STORE BY HERO

Home Catalog Men's Women's T Shirts

The Apparel Store

Name	Status	Type	Initiator	Size	Time	Waterfall
loader.js	200	script	(index):933	23.9...	17...	
display?appid=De...	200	xhr	shop_eye...	659 B	66...	

2 / 22 requests | 24.5 kB / 529 kB transferred | 90.1 kB / 1.2 MB resources | Finish: 1.13 s | DOMCo

Incorrect result

- ✗ loader.js request is not displaying in the Network log
- ✗ Status column does not display 200 as the HTTP response code

Are Dashboard configurations causing a display issue?

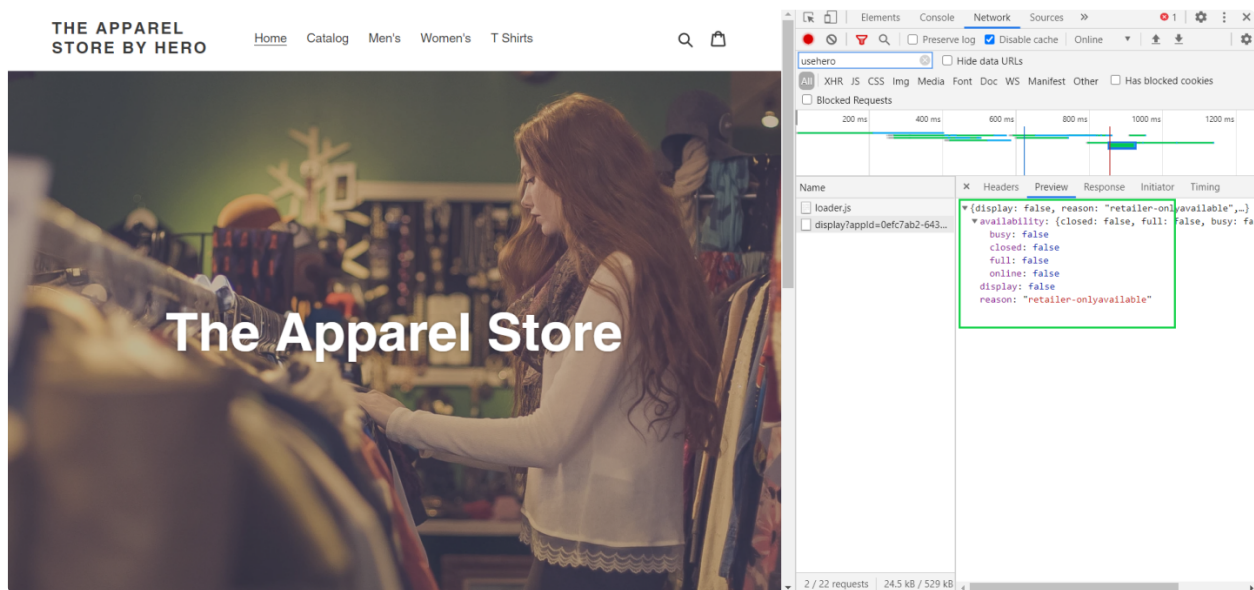
Once you have correctly deployed Virtual Shopping to your website, you may find that On-site Chat is still not displaying as a result of configurations set up for you in your Dashboard, such as Store Hours, Only Available, Get Notified 24/7 and others.

To help you find out if a configuration is causing the issue, you can check Display Request responses. Below we've put together common Display Request responses to help resolve issues quickly.

Common Display Request responses

Retailer-only available

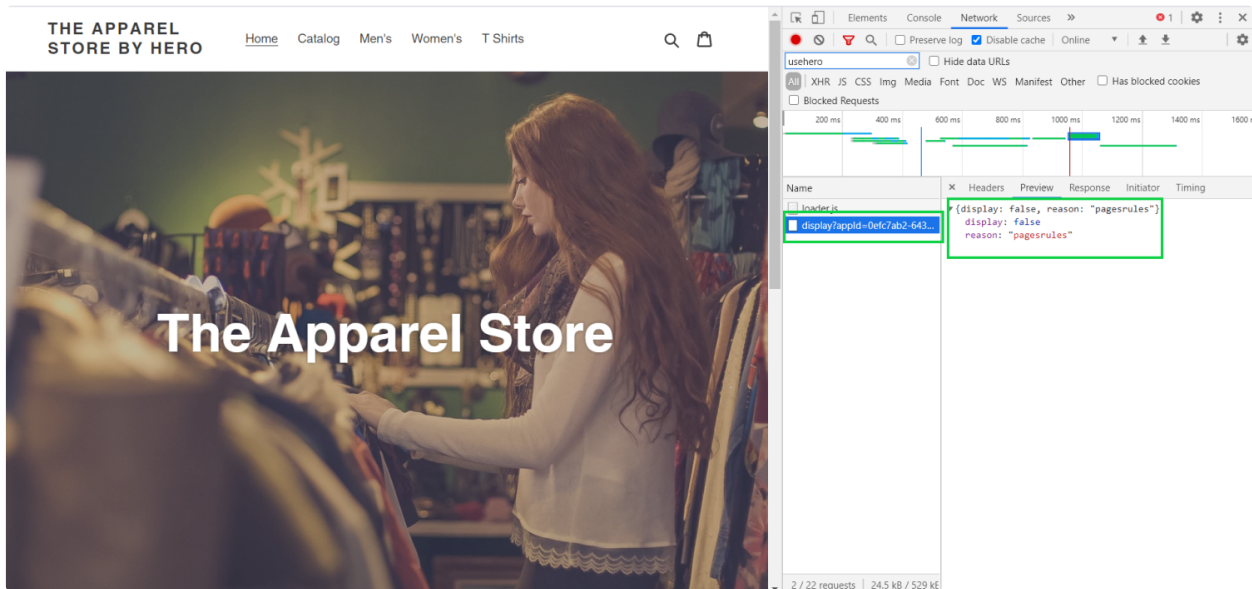
A display response of `retailer-onlyavailable` simply means that your plugin configurations in Dashboard, has been switched from On to Only available. Therefore, **On-site Chat will only display when a team member is online and available on the Store app.** If store teams are offline or busy then the Launcher Icon will not display.



Retailer-only available

Page rules

A display response of `pagerules` means that the On-site Chat is hidden on the current page because of [blacklisting or whitelisting](#), also configurable in your Dashboard. **If the page has been blacklisted, then the plugin will only display on that page under the conditions of the follow rules.**

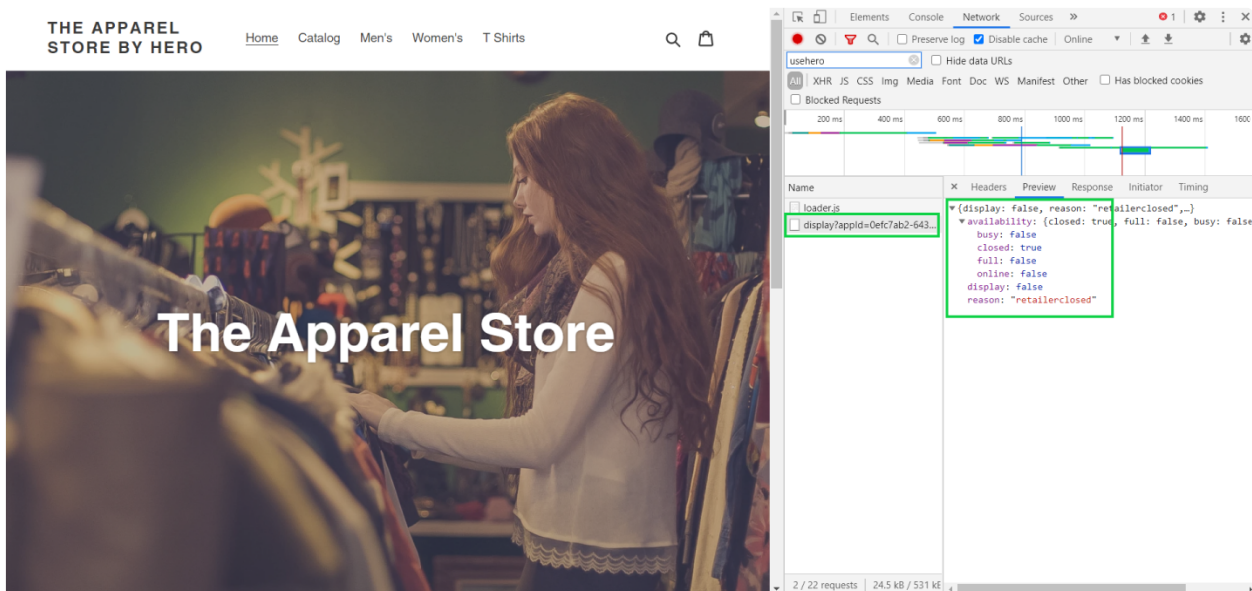


Page rules

Retailer closed

A display response of `retailerclosed` means that On-site Chat is not displaying due to the store currently being closed.

This will be based on the store hours that have been set in the dashboard. **This response will only show if On-site Chat visibility is set to show within Store Hours.** Only Available and Get Notified 24/7 are disabled.



Retailer closed

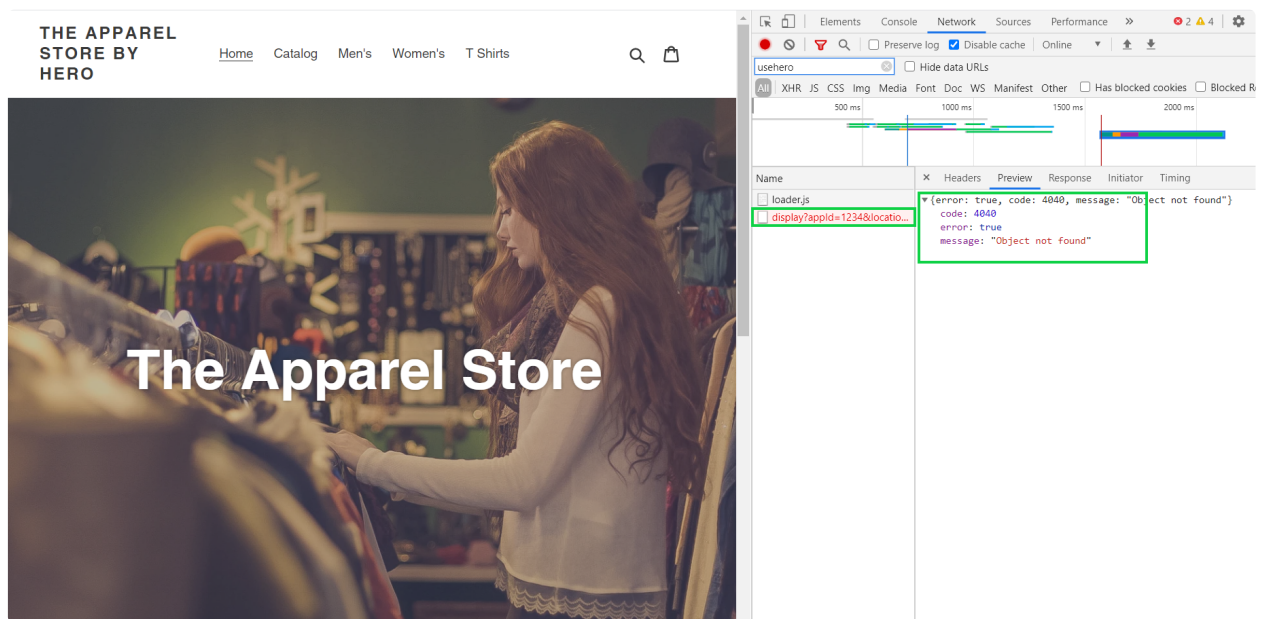
Retailer-full

A display response of `Retailerfull` means that **On-site Chat is currently not showing because you have reached your limit of Get Notified lead chats in the queue.**

This limit is typically set at 30 but you can easily change this in your dashboard. On-site chat should display again once the number of chats has gone below the lead queue limit once store teams start accepting the backlog of chats where customers have left a message.

Object not found - 4040 Error message

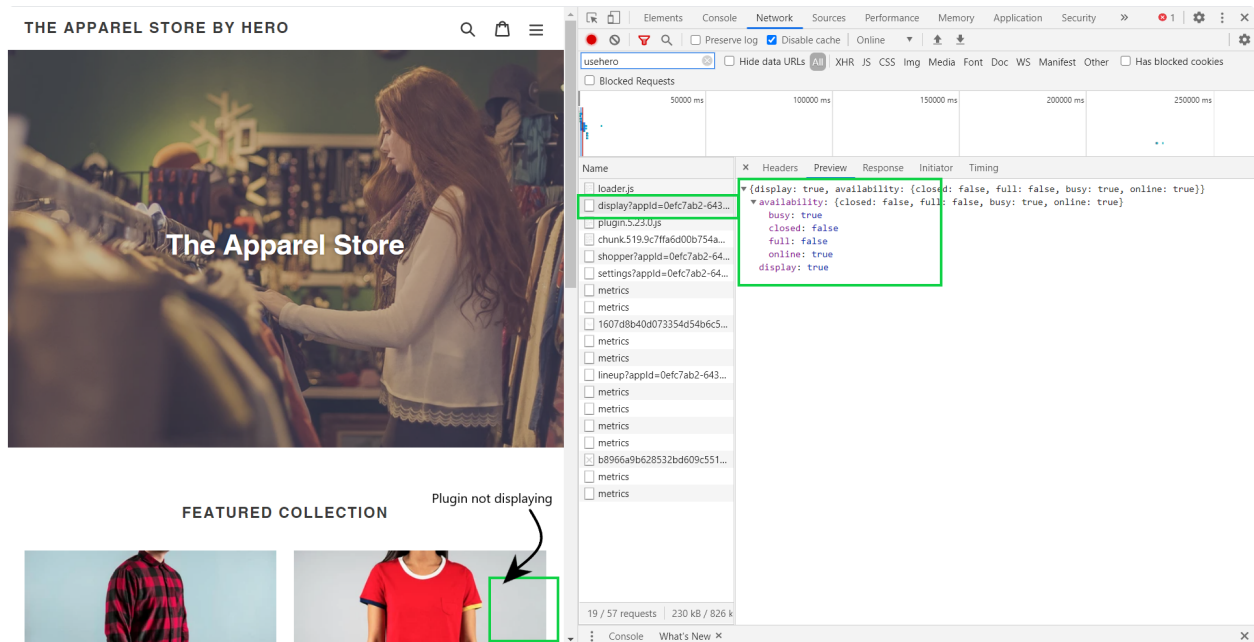
A display response of `Object not found` means that the **On-site chat is not currently displaying on your webpages because an incorrect App ID has been deployed in the JavaScript tag**. To avoid this error, ensure that the right App ID has been implemented correctly across your web pages.



Object not found - 4040 Error message

Is On-site chat being hidden by CSS?

If the Display Request for the On-site is 'true' but it is still not displaying, this may be because the On-site chat element `hero-iframe-container` has been hidden with CSS using the CSS property: `display:none`.



Steps to test

1. Right-click to inspect page
2. Ensure you are viewing the Elements tab
3. Search and select the element div class: `hero-launcher-avatar-container-active-standard-animating`
4. View the Styles tab on the Elements panel to view the CSS rules being applied to the On-site chat element.
5. If in the styles tab you see the `display:none` this will be why On-site chat is not showing up on your webpage.

THE APPAREL STORE BY HERO

The Apparel Store

FEATURED COLLECTION

```
<div class="hero-launcher-avatar-container active standard animating" style="display: none; width: 60px; height: 60px;">
  <iframe data-qa="launcher-icon-iframe" class="hero-launcher-avatar" scrolling="no"></iframe>
  <div class="hero-launcher-push-container inactive" style="width: 292px; height: 168px;"></div>
  <div class="hero-plugin-container inactive"></div>
  <iframe class="hero-media"></iframe>
</div>
```

```
display: none;
width: 60px;
height: 60px;
```

Here, Virtual Shopping Messenger is not visible on the webpage because it being hidden with CSS.

Test Shopper Event tracking

Sometimes after setting up Shopper Events, they do not behave as expected. There are a number of reasons why this may happen so to help discover issues and fix them, we've put together this section where we'll walk you through:

- ✔ **How to test the set up of Shopper Event tracking**
- ✔ **What you should expect to see on your Staging or Production environment**
- ✔ **How to test events using the Store app and developer tools on your browser**



How can I test-run the experience?



Is Shopper Event tracking deployed correctly?



Are Shopper Events firing correctly?



Troubleshooting

How can I test-run the experience?

The best way to check that your Shopper Event tracking is correctly setup is to test run the experience a store team member would have when messaging with a shopper through the Store App.

Testing out the interaction itself will help you check that Shopper Events are tracking as expected and surfacing insights to store teams. To do this, you'll need to create a Store App account and claim a chat to try out the experience.

Before you test

To ensure you can **create a Store App account**, you will need to:

- Store location has been added to your Dashboard.
- At least one member of your admin or management team has access to manage teams in the Dashboard so they can approve pending member accounts

To ensure **you are configured for the test to go to your Dashboard**, you will need to make sure the following is true:

- "Only Available" is off
- Ensure "Get Notified 24/7" is on

How to run a test using the Store app

1. Download the Store app



Klarna | Store App for Teams 4+

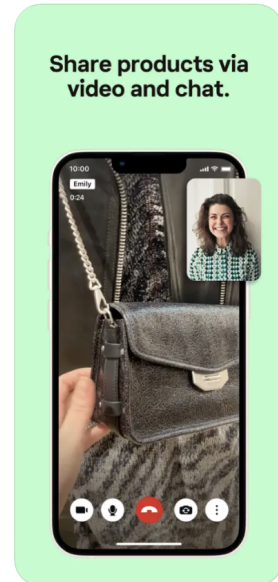
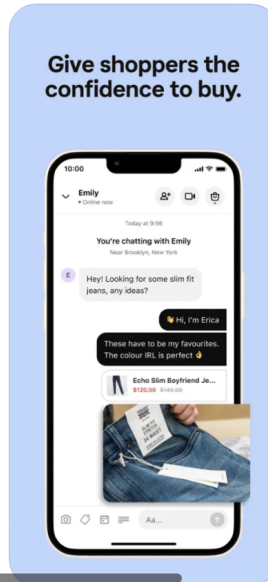
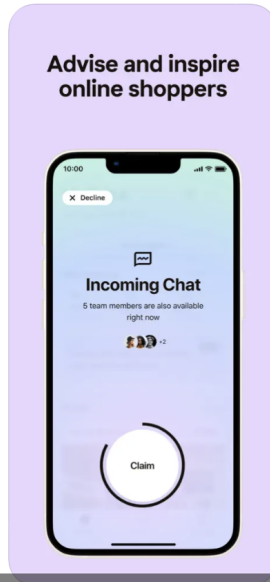
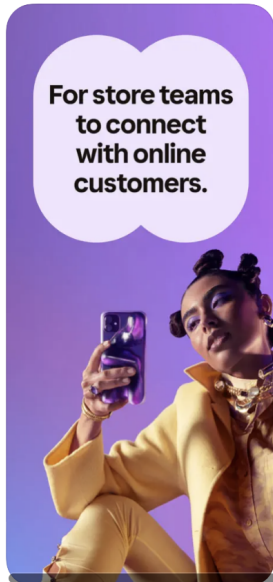
Team app for Virtual Shopping

[Klarna Bank AB](#)

★ ★ ★ ★ 2.5 • 2 Ratings

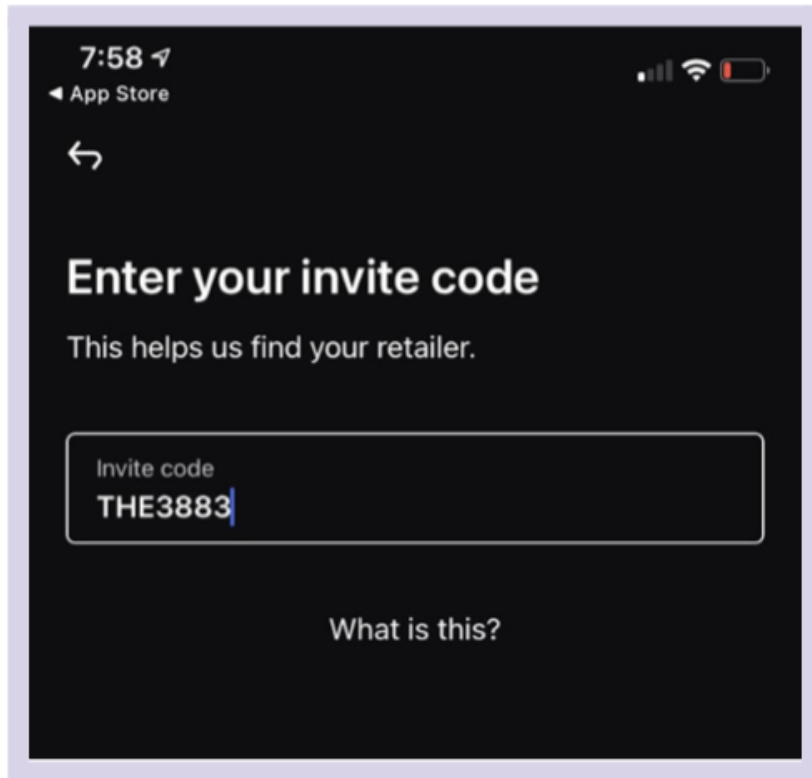
Free

Screenshots [iPhone](#) [iPad](#)



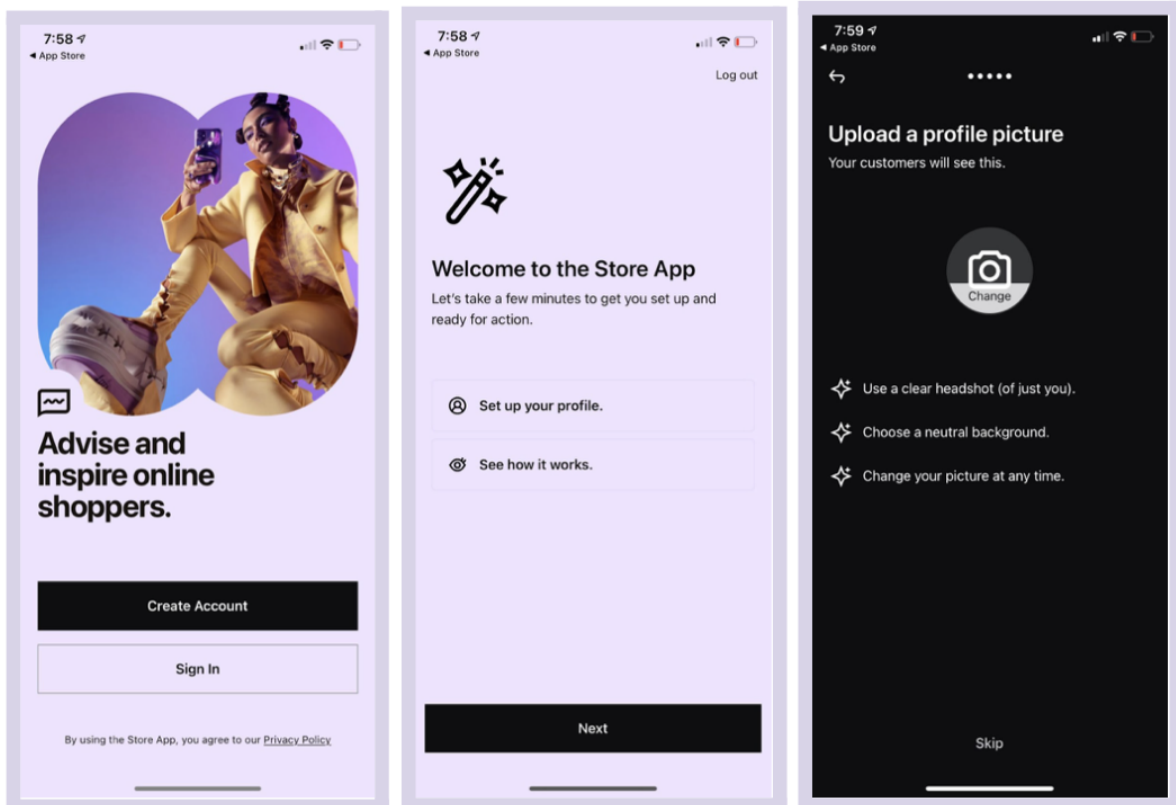
Also available in Google Play for Android

2. Create an account using the seven-character invite code. This code - a Staging code and Production code - will have been sent to you along with your unique Staging and Production App IDs

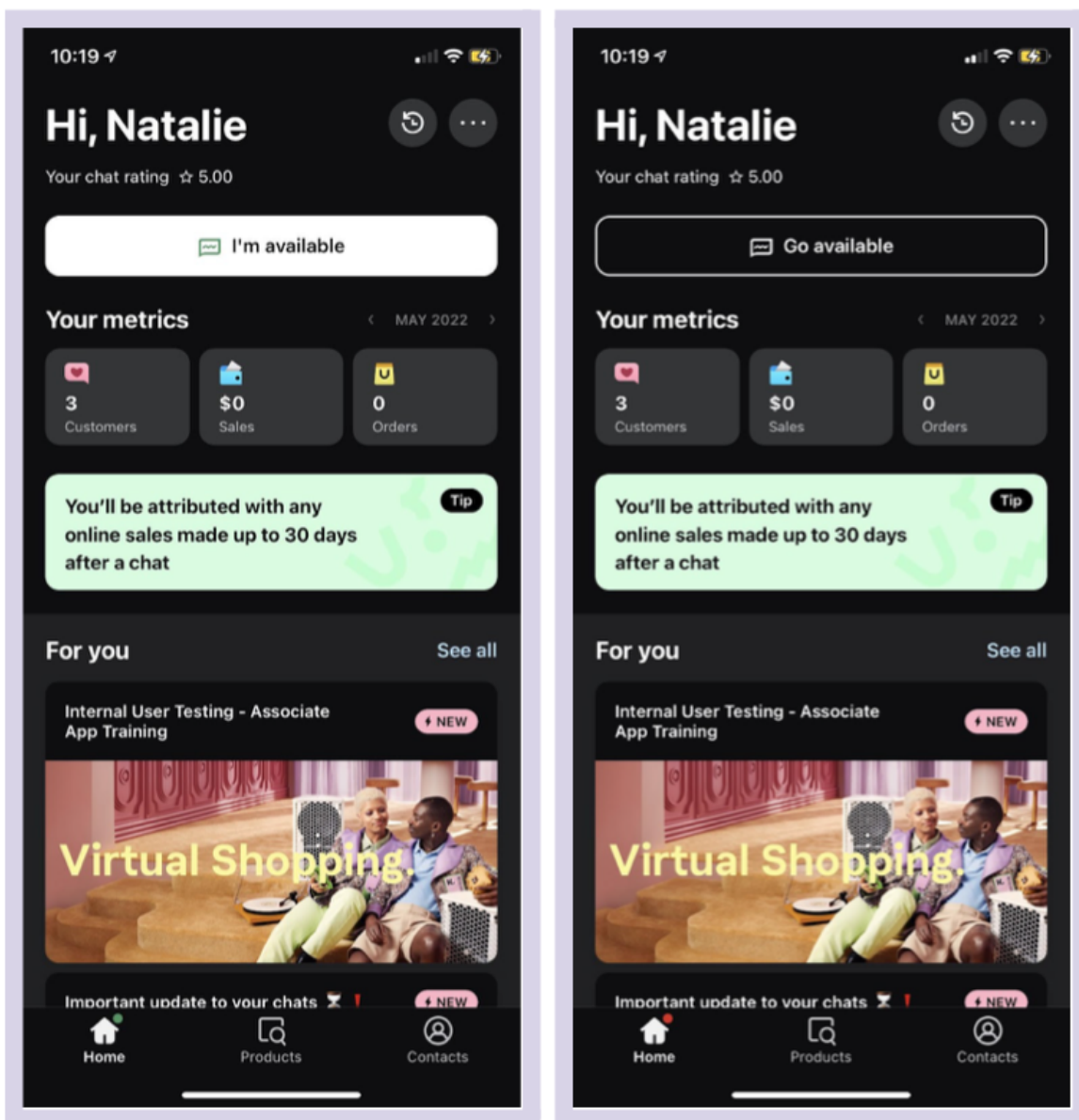


Enter your invite code to create your account

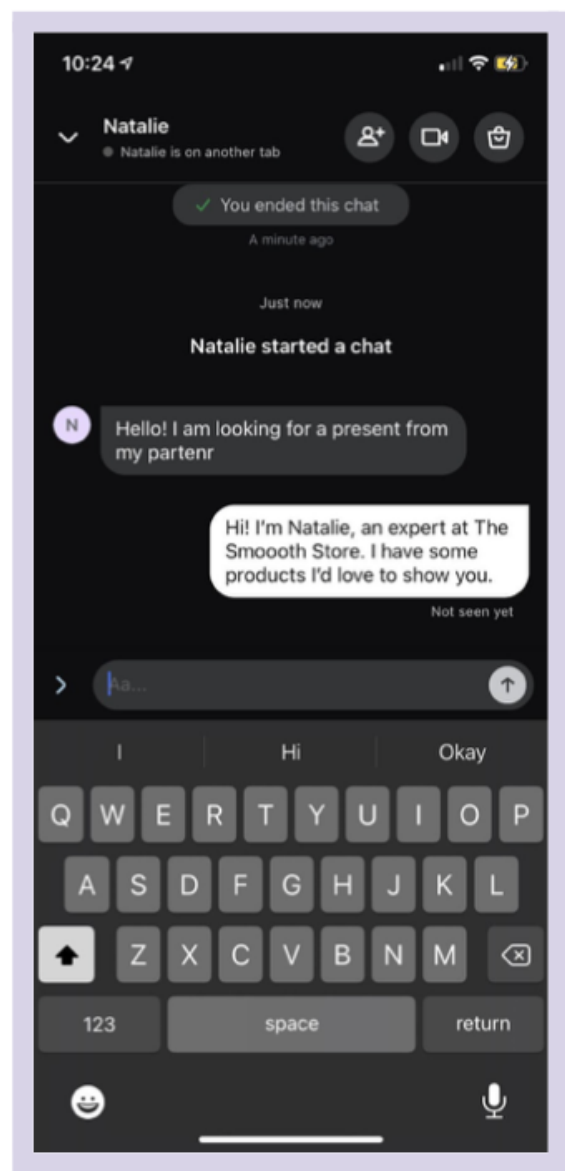
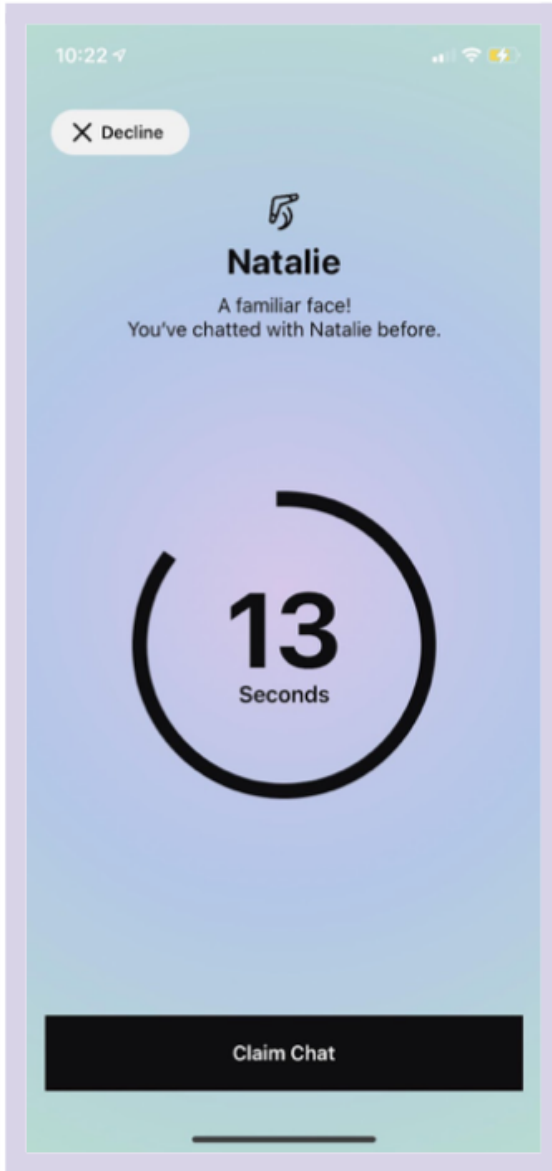
3. To register, you will first need to go through the onboarding process which includes creating a profile and an introduction to Virtual Shopping.



- Once you have completed all three onboarding steps, a member of your management or admin team will need to approve the account in the Dashboard.
- Once your account has been approved, you will then need to make yourself available on the Store app by tapping on the **Go Available** button



- Start a chat using the On-site Chat feature and accept the chat on the Store App.



Is Shopper Event tracking deployed correctly?

What success looks like

To help you ensure that Shopper Event tracking is correctly set-up, we've provided guidelines and examples of successful implementations:



Product View



Purchase



Basket Add



Basket Remove



Category View



Search

Product View

Are Shopper Events deployed correctly?

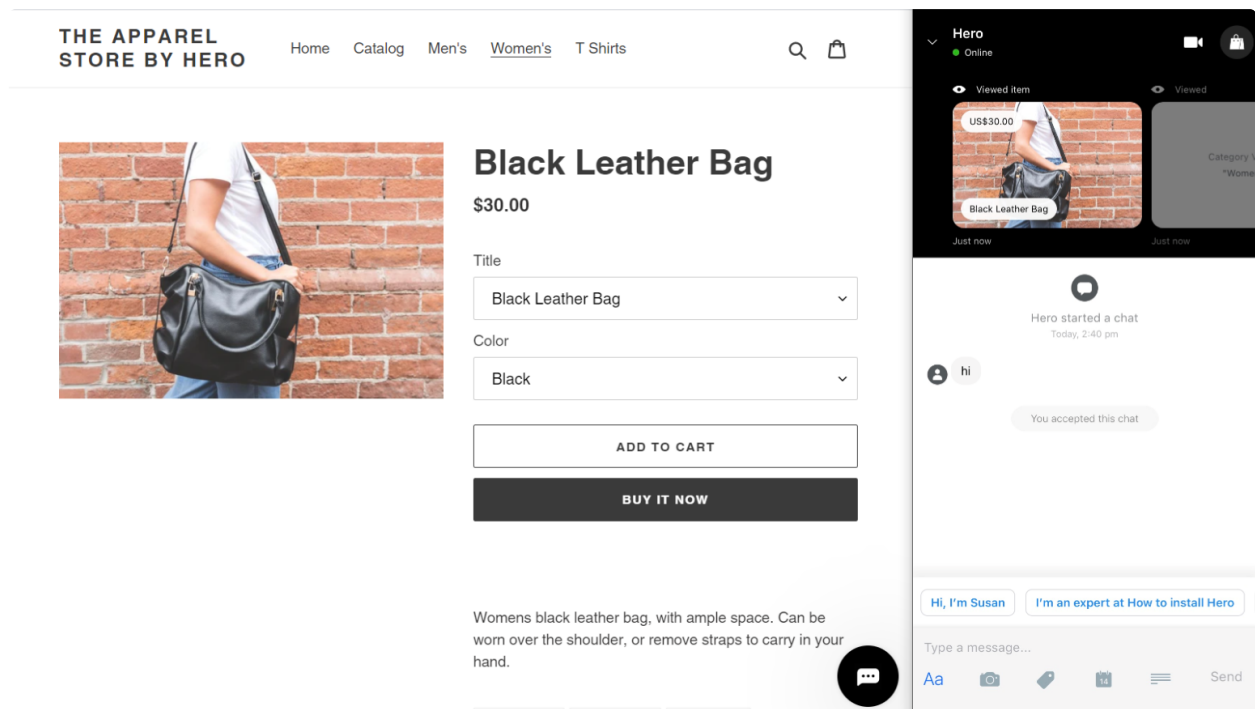
Below is an example of what a set-up looks like when the 'Product View' Shopper Event has been added correctly to the product detail pages of your website. Before you begin comparing your setup, make sure that you have completed the following:

⚠ Before you begin

1. To ensure you have deployed this Shopper Event correctly make sure you have included all the correct [fields](#).
2. Follow the steps in the ['How can I test-run the experience'](#) section to ensure that the correct details are being passed in the event and captured on the Store App as expected.

What success looks like

1. With a chat in progress, the 'Product View' event is displayed on the Store App.



2. Here the 'Product View' event can be seen in the request payload



Black Leather Bag

\$30.00

Title
Black Leather Bag

Color
Black

ADD TO CART

The screenshot shows the Network tab in Chrome DevTools. The selected request is a tracking event. The payload is highlighted in green and contains the following JSON data:

```
{
  "applicationId": "0efc7ab2-6433-4612-b6fc-06e9d993886c",
  "applicationId": "0efc7ab2-6433-4612-b6fc-06e9d993886c",
  "events": [
    {
      "id": "7dc13c82-48a1-44a1-8503-2cce454f69db",
      "date": "2020-12-14T09:38:30.554Z",
      "id": "7dc13c82-48a1-44a1-8503-2cce454f69db"
    },
    {
      "id": "1606771592",
      "name": "Black Leather Bag",
      "brand": "Hero",
      "category": "Bag",
      "currency": "USD",
      "id": "1606771592",
      "image": "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/black-bag-over-the-shoulder_925x_f308436f-9e65-4754-897c-3c8bf631a5b4_925x.jpg?v=1606771592",
      "location": "https://the-apparel-store-by-hero.myshopify.com/products/black-leather-bag",
      "name": "Black Leather Bag",
      "price": 30,
      "variant": "Black Leather Bag / Black",
      "type": "ecommerce:detail"
    }
  ]
}
```

3. Here, the 'Product View' event can be seen in the Network log of the request payload.

The screenshot shows the Network tab in Chrome DevTools on the left and a mobile app chat interface on the right. The Network tab shows a request payload with a 'Product View' event highlighted in green. The chat interface shows a message from 'Hero' with a product image and price, and a response from 'Susan'.

The Network tab payload is the same as in the previous screenshot, but the 'Product View' event is highlighted in green.

The chat interface shows a message from 'Hero' with a product image and price, and a response from 'Susan'.



Group IDs vs. Variant IDs

If your Product Feed includes both Group IDs and Variant IDs, then it is important to ensure that you are providing the correct ID in the Shopper Event tracking to match the correct ID of the product being tracked.

This will ensure that product variants are also captured on the app when a shopper views or purchases a specific product variant.

For example:

- If a shopper lands on a generic product page - assuming the shopper has not selected a variant i.e size, colour, length - then the Group ID for that product should be passed as the product ID for the Shopper Event.
- If a shopper selects a variant of that product and the page reloads to display the variant product, then the Variant ID should be passed as the product ID for the Shopper Event.

Purchase

Are Shopper Events deployed correctly?

Below is an example of what a set-up looks like when the 'Purchase' Shopper Event has been added correctly to the order confirmation pages of your website". Before you begin comparing your set-up, make sure that you have completed the following:

⚠ Before you begin

1. To ensure you have deployed this Shopper Event correctly make sure you are using the correct [fields](#).
2. Follow the steps in the ['How can I test-run the experience'](#) section to ensure that the correct details are being passed in the event and captured on the Store App as expected.

What success looks like

1. Here, you can see the Purchase event in the request payload of the Network tab

The screenshot displays a web browser window with two main sections. On the left, the 'The Apparel Store by Hero' order confirmation page is visible. It shows a 'Hide order summary' link, a subtotal of \$40.00 (highlighted with a green box), and a total of \$53.54. Below the order summary, there is a 'Thank you margaret!' message and a map showing the shipping address in London. On the right, the browser's network tab is open, showing a request payload for a purchase event. The payload includes an application ID, a list of events with timestamps, a list of products (including 'Chequered Red Shirt' with brand 'Hero' and price 40), and a purchase object with details like order ID '1038', shipping cost '13.54', and subtotal '40'. The purchase object is also highlighted with a green box.

2. On the right, you can see the Purchase event is displaying correctly in the Store app

```
applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c"
events: [{"id": "37851dae-b340-4d5f-874b-7174e50fa06d", date: "2020-12-17T12:49:00.718Z", ...}
        {id: "37851dae-b340-4d5f-874b-7174e50fa06d", date: "2020-12-17T12:49:00.718Z", ...}
        {id: "37851dae-b340-4d5f-874b-7174e50fa06d", date: "2020-12-17T12:49:00.718Z", ...}
products: [{"id": "1606771598", quantity: 1, name: "Chequered Red Shirt", ...}
            {id: "1606771598", quantity: 1, name: "Chequered Red Shirt", ...}
purchase: {currency: "USD", id: "1038", total: 53.54, shippingCost: 13.54, ...}
coupon: "PROXD10"
currency: "USD"
id: "1038"
shippingCost: 13.54
shippingMethod: "First Class Package International"
subtotal: 40
tax: 0
total: 53.54
type: "ecommerce:purchase"
```

Group IDs vs. Variant IDs

If your Product Feed includes both Group IDs and Variant IDs, then it is important to ensure that you are providing the correct ID in the Shopper Event tracking to match the correct ID of the product being tracked.

This will ensure that product variants are also captured on the app when a shopper views or purchases a specific product variant.

For example:

- If a shopper lands on a generic product page - assuming the shopper has not selected a variant i.e size, colour, length - then the Group ID for that product should be passed as the product ID for the Shopper Event.
- If a shopper selects a variant of that product and the page reloads to display the variant product, then the Variant ID should be passed as the product ID for the Shopper Event.

Basket Add

Are Shopper Events deployed correctly?

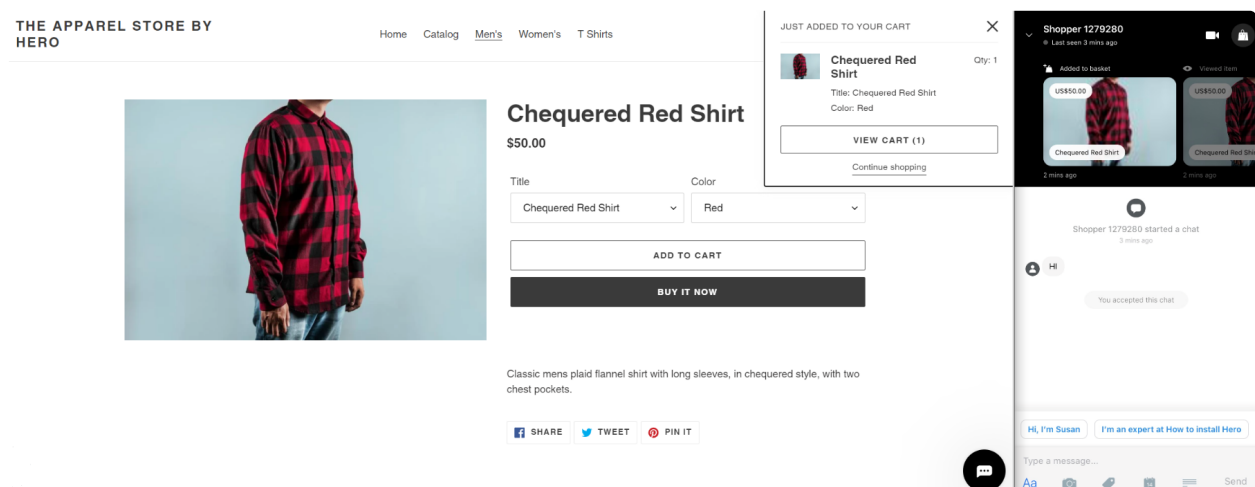
Below is an example of what a set-up looks like when the 'Basket Add' Shopper Event has been added correctly to the product detail pages of your website. Before you begin comparing your set-up, make sure that you have completed the following:

⚠ Before you begin

1. To ensure you have deployed this Shopper Event correctly make sure you are using the correct [subfields](#).
2. Follow the steps in the ['How can I test-run the experience'](#) section to ensure that the correct details are being passed in the event and captured on the Store App as expected.

What success looks like

1. With a chat in progress, the 'Basket Add' event is captured on the Store app



2. Here, the 'Basket Add' event can be seen in the request payload

JUST ADDED TO YOUR CART

Chequered Red Shirt Qty: 1
Title: Chequered Red Shirt
Color: Red

VIEW CART (1)
Continue shopping

Chequered Red Shirt
\$50.00

Title: Chequered Red Shirt
Color: Red

ADD TO CART
BUY IT NOW

```

Name: tracking
Response Headers (18)
Request Headers (18)
Request Payload
{
  applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c",
  events: [...],
  0: {
    id: "2794f22e-a3a0-4ef9-86b3-e6705ba2f099",
    date: "2020-12-14T07:29:48.502Z",
    type: "ecommerce:add",
    products: [
      {
        id: "1606771598",
        quantity: 1,
        basketQuantity: 1,
        name: "Chequered Red Shirt",
        basketQuantity: 1,
        brand: "Hero",
        category: "Shirt",
        currency: "USD",
        id: "1606771598",
        image: "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/red-plaid-shirt_925x_b9b019f9-01a5-4515-89b0-ae6da80f0671_small.jpg?v=1606771598",
        location: "https://the-apparel-store-by-hero.myshopify.com/products/chequered-red-shirt",
        name: "Chequered Red Shirt",
        price: 50,
        quantity: 1,
        variant: "Chequered Red Shirt / Red",
        type: "ecommerce:add"
      }
    ]
  }
}

```

3. On the right, the 'Basket Add' event can be seen in the Store app

Shopper 1279280
Last seen 3 mins ago

Added to basket
US\$50.00
Chequered Red Shirt

Viewed item
US\$50.00
Chequered Red Shirt

Shopper 1279280 started a chat
3 mins ago

Hi

You accepted this chat

Hi, I'm Susan | I'm an expert at How to install Hero

Type a message...

```

Name: tracking
Response Headers (18)
Request Headers (18)
Request Payload
{
  applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c",
  events: [...],
  0: {
    id: "75fc0f27-0956-4ebb-bcdd-c0b939820028",
    date: "2020-12-14T06:47:52.563Z",
    type: "ecommerce:add",
    products: [
      {
        id: "1606771598",
        quantity: 1,
        basketQuantity: 1,
        name: "Chequered Red Shirt",
        basketQuantity: 1,
        brand: "Hero",
        category: "Shirt",
        currency: "USD",
        id: "1606771598",
        image: "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/red-plaid-shirt_925x_b9b019f9-01a5-4515-89b0-ae6da80f0671_small.jpg?v=1606771598",
        location: "https://the-apparel-store-by-hero.myshopify.com/products/chequered-red-shirt",
        name: "Chequered Red Shirt",
        price: 50,
        quantity: 1,
        variant: "Chequered Red Shirt / Red",
        type: "ecommerce:add"
      }
    ]
  }
}

```



Group IDs vs. Variant IDs

If your Product Feed includes both Group IDs and Variant IDs, then it is important to ensure that you are providing the correct ID in the Shopper Event tracking to match the correct ID of the product being tracked.

This will ensure that product variants are also captured on the app when a shopper views or purchases a specific product variant.

For example:

- If a shopper lands on a generic product page - assuming the shopper has not selected a variant i.e size, colour, length - then the Group ID for that product should be passed as the product ID for the Shopper Event.
- If a shopper selects a variant of that product and the page reloads to display the variant product, then the Variant ID should be passed as the product ID for the Shopper Event.

Basket Remove

Are Shopper Events deployed correctly?

Below is an example of what a set-up looks like when the 'Basket Remove' Shopper Event has been added correctly to the product detail pages of your website. Before you begin comparing your set-up, make sure that you have completed the following:

⚠ Before you begin

1. To ensure you have deployed this Shopper Event correctly make sure you are using the correct [fields](#).
2. Follow the steps in the ['How can I test-run the experience'](#) section to ensure that the correct details are being passed in the event and captured on the Store App as expected.

What success looks like

1. Here, you can see the 'Basket Remove' event in the request payload of the Network tab.

The screenshot shows a web browser with the 'THE APPAREL STORE BY HERO' website. The left sidebar displays 'Your cart' with the message 'Your cart is currently empty.' and a 'CONTINUE SHOPPING' button. Below the cart are 'Quick links' and a search bar. The main content area shows a product detail page for a 'Red Sports Tee'. The right side of the browser shows the 'Network' tab with a request payload for a 'Basket Remove' event. The event data is highlighted in a green box and includes the following details:

```
{
  "applicationId": "0efc7ab2-6433-4612-b6fc-06e9d993886c",
  "applicationId": "0efc7ab2-6433-4612-b6fc-06e9d993886c",
  "events": [
    {
      "id": "69af4c5b-a753-4988-806e-d82656f943b7",
      "date": "2020-12-17T10:47:41.223Z",
      "id": "69af4c5b-a753-4988-806e-d82656f943b7"
    },
    {
      "products": [
        {
          "id": "1606771604",
          "quantity": 1,
          "basketQuantity": 0,
          "name": "Red Sports Tee",
          "basketQuantity": 0,
          "brand": "Hero",
          "category": "Top",
          "currency": "USD",
          "id": "1606771604",
          "image": "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/womens-red-t-shirt_925x_b78d289c-e4d9-4b00-bde3-671c43b03041.jpg?v=1606771604",
          "location": "https://the-apparel-store-by-hero.myshopify.com/products/red-sports-tee?variant=37306687340189",
          "name": "Red Sports Tee",
          "price": 50,
          "quantity": 1,
          "variant": "Red Sports Tee / Red",
          "type": "ecommerce:remove"
        }
      ]
    }
  ]
}
```

2. On the right, the 'Basket Remove' event is showing correctly on the Store app

The image shows a browser's developer tools network tab on the left and a mobile app chat interface on the right. The network tab displays a tracking request with a status code of 200. The request payload is highlighted with a green box and contains the following JSON data:

```

{
  "applicationId": "0efc7ab2-6433-4612-b6fc-06e9d993886c",
  "applicationId": "0efc7ab2-6433-4612-b6fc-06e9d993886c",
  "events": [
    [
      {
        "id": "69af4c5b-a753-4988-806e-d82656f943b7",
        "date": "2020-12-17T10:47:41.223Z",
        "id": "69af4c5b-a753-4988-806e-d82656f943b7"
      },
      {
        "id": "69af4c5b-a753-4988-806e-d82656f943b7"
      }
    ],
    [
      {
        "id": "1606771604",
        "quantity": 1,
        "basketQuantity": 0,
        "name": "Red Sports Tee",
        "basketQuantity": 0,
        "brand": "Hero",
        "category": "Top",
        "currency": "USD",
        "id": "1606771604",
        "image": "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/womens-red-t-shirt_925x_b78d289c-e4d9-4b00-bde3-671c43b03841.jpg?v=1606771604",
        "location": "https://the-apparel-store-by-hero.myshopify.com/products/red-sports-tee?variant=37306887340189",
        "name": "Red Sports Tee",
        "price": 50,
        "quantity": 1,
        "variant": "Red Sports Tee / Red",
        "type": "ecommerce:remove"
      }
    ]
  ]
}

```

The mobile app chat interface on the right shows a conversation with a user named 'Shopper 1290556'. The chat history includes a message 'hi' and a confirmation 'You accepted this chat'. At the bottom, there are buttons for 'Hi, I'm Margaret' and 'I'm an expert at How to install Hero', and a text input field for sending a message.

i Group IDs vs. Variant IDs

If your Product Feed includes both Group IDs and Variant IDs, then it is important to ensure that you are providing the correct ID in the Shopper Event tracking to match the correct ID of the product being tracked.

This will ensure that product variants are also captured on the app when a shopper views or purchases a specific product variant.

For example:

- If a shopper lands on a generic product page - assuming the shopper has not selected a variant i.e size, colour, length - then the Group ID for that product should be passed as the product ID for the Shopper Event.
- If a shopper selects a variant of that product and the page reloads to display the variant product, then the Variant ID should be passed as the product ID for the Shopper Event.

Category View

Are Shopper Events deployed correctly?

Below is an example of what a set-up looks like when the 'Category View' Shopper Event has been added correctly to the product detail pages of your website. Before you begin comparing your set-up, make sure that you have completed the following:

⚠ Before you begin

1. To ensure you have deployed this Shopper Event correctly make sure you are using the correct [fields](#).
2. Follow the steps in the ['How can I test-run the experience'](#) section to ensure that the correct details are being passed in the event and captured on the Store App as expected.

What success looks like

1. Here, you can see the 'Category View' event displayed in the request payload of the Network tab

The screenshot shows a web browser displaying a product category page for 'Mens' at 'THE APPAREL STORE BY HERO'. The page features a search bar, a filter dropdown set to 'All products', and a sort dropdown set to 'Best selling'. Below the filters, there are six product cards. The first two are 'Chequered Red Shirt \$50.00' and 'Ocean Blue Shirt \$50.00'. The browser's developer tools are open to the Network tab, showing a request to 'https://api.usehero.com/webplugin/tracking'. The request method is POST, and the status code is 200. The request payload is expanded, showing a JSON object with an 'events' array. One event is highlighted with a green box, containing the following details: 'action': 'category-view', 'date': '2020-12-11T09:51:40.034Z', 'id': 'faaa3389-7a98-46f5-b02a-4ec5f8739147', 'location': 'https://the-apparel-store-by-hero.myshopify.com/collections/mens', 'type': 'event', and 'value': 'Mens'.

2. On the right, you can see the 'Category View' event, showing correctly in the Store app

Elements Console Network Sources Performance Memory Application Security Lighthouse

tracking

Blocked Requests

2000 ms 4000 ms 6000 ms 8000 ms 10000 ms 12000 ms 14000 ms

Name consent-tracking-api... tracking tracking

Request URL: <https://api.usehero.com/webplugin/tracking>
 Request Method: POST
 Status Code: 200
 Remote Address: 54.76.69.248:443
 Referrer Policy: strict-origin-when-cross-origin

Response Headers (18)
 Request Headers (18)
 Request Payload view source

```

{applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c",...}
  applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c"
  events: [{id: "4af502b1-4a2f-408f-806c-cd1420dd2896", date: "2020-12-11T09:25:02.615Z", type: "event",...}]
  0: {id: "4af502b1-4a2f-408f-806c-cd1420dd2896", date: "2020-12-11T09:25:02.615Z", type: "event",...}
    action: "category-view"
    date: "2020-12-11T09:25:02.615Z"
    id: "4af502b1-4a2f-408f-806c-cd1420dd2896"
    location: "https://the-apparel-store-by-hero.myshopify.com/collections/mens"
    type: "event"
    value: "Mens"
  
```

Shopper 1271231
 London, GB

Viewed

Category Viewed "Mens"

US\$50.00

Ocean Blue Shirt

A minute ago

Shopper 1271231 started a chat Just now

Hi!

You accepted this chat

Hi, I'm Susan I'm an expert at How to install Hero

Type a message...

Aa Send

Search

Are Shopper Events deployed correctly?

Below is an example of what a set-up looks like when the 'Search' Shopper Event has been added correctly to the product detail pages of your website. Before you begin comparing your set-up, make sure that you have completed the following:

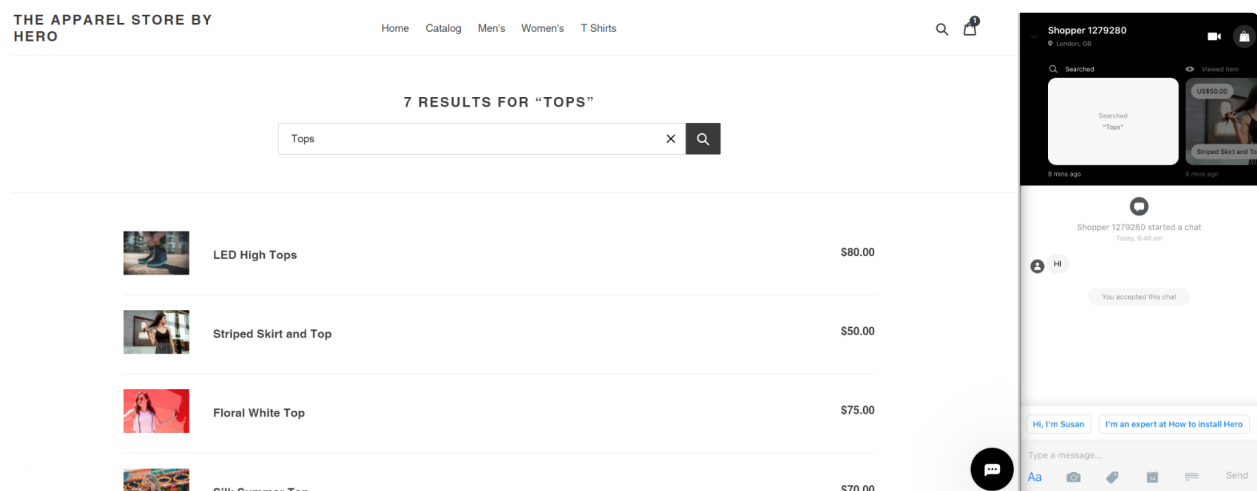


Before you begin

1. To ensure you have deployed this Shopper Event correctly make sure you are using the correct [fields](#).
2. Follow the steps in the ['How can I test-run the experience'](#) section to ensure that the correct details are being passed in the event and captured on the Store App as expected.

What success looks like

1. With a chat in progress, the 'Search' Shopper Event is displayed in the Store App.



2. Here, you can see the 'Search' event in the request payload of the Network tab

THE APPAREL STORE BY HERO

7 RESULTS FOR "TOPS"

Tops

- LED High Tops \$80.00
- Striped Skirt and Top \$50.00
- Floral White Top \$75.00
- Silk Summer Top \$70.00
- Long Sleeve Cotton Top \$50.00
- Classic Varsity Top

Network Request Details:

- Request URL: `https://api.usehero.com/webplugin/tracking`
- Request Method: POST
- Status Code: 200
- Request Payload:


```
{
    "applicationId": "0efc7ab2-6433-4612-b6fc-06e9d993886c",
    "events": [
      {
        "id": "ca227d08-f7a9-4cb6-8b15-257b18edb12f",
        "date": "2020-12-11T10:26:51.467Z",
        "type": "event",
        "action": "search",
        "date": "2020-12-11T10:26:51.467Z",
        "id": "ca227d08-f7a9-4cb6-8b15-257b18edb12f",
        "location": "https://the-apparel-store-by-hero.myshopify.com/search?q=Tops&type="
      }
    ]
  }
```

3. On the right, you can see the 'Search' Shopper Event is displaying correctly in the Store App.

Shopper 1271231

Online

Searched "Tops"

Viewed Item

US\$75.00

Floral White Top

You accepted this chat

You finished this chat 48 mins ago

Shopper 1271231 started a chat

hi

You accepted this chat

Hi, I'm Susan | I'm an expert at How to install Hero

Type a message...

Network Request Details:

- Request URL: `https://api.usehero.com/webplugin/tracking`
- Request Method: POST
- Status Code: 200
- Request Payload:


```
{
    "applicationId": "0efc7ab2-6433-4612-b6fc-06e9d993886c",
    "events": [
      {
        "id": "ca227d08-f7a9-4cb6-8b15-257b18edb12f",
        "date": "2020-12-11T10:26:51.467Z",
        "type": "event",
        "action": "search",
        "date": "2020-12-11T10:26:51.467Z",
        "id": "ca227d08-f7a9-4cb6-8b15-257b18edb12f",
        "location": "https://the-apparel-store-by-hero.myshopify.com/search?q=Tops&type="
      }
    ]
  }
```

Are Shopper Events firing correctly?

Once you have set up the Shopper Events on your pages, you will need to check the events are firing as expected. To help you do this, we've put together the following test scenarios.

To check that events are firing as expected, you will need go to your Staging or Production environment, start a Virtual Shopping chat and view the developer tools request payload.

Get ready to test

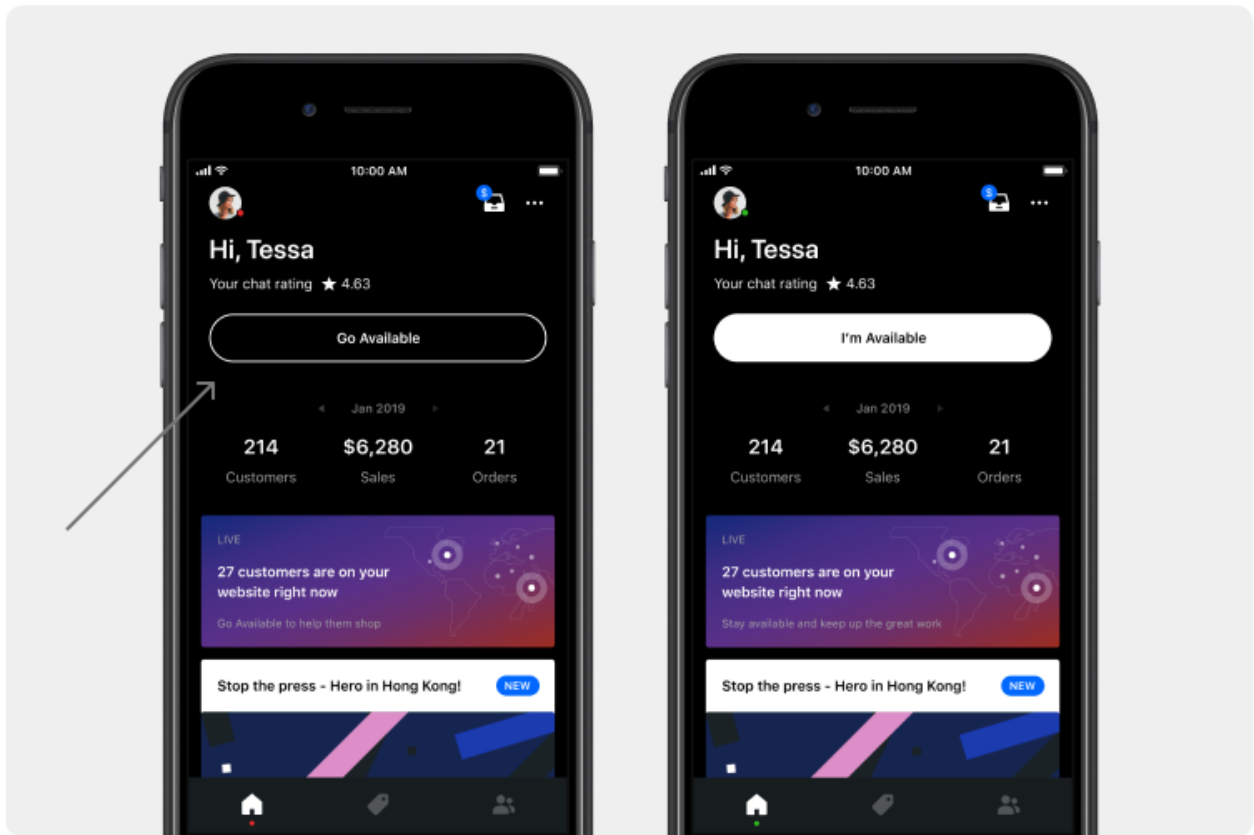
Before you begin, you need to complete the following 2 preliminary steps:

1. Javascript successfully loaded on page and Virtual Shopping On-site Chat is displaying as expected

The screenshot shows a web browser displaying the 'The Apparel Store' website. The page features a navigation menu with 'Home', 'Catalog', 'Men's', and 'Women's' options, and a search icon. A chat widget is visible in the bottom right corner, with the text 'Speak to our in-store team' and a 'Ask a question' input field. A network developer tool is open on the right side of the browser, showing a list of network requests. The 'loader.js' request is highlighted in green, indicating it has been successfully loaded. The network tool also shows various other requests, including 'shop_events.js' and 'metrics', which are likely related to the Shopper Events being tested.

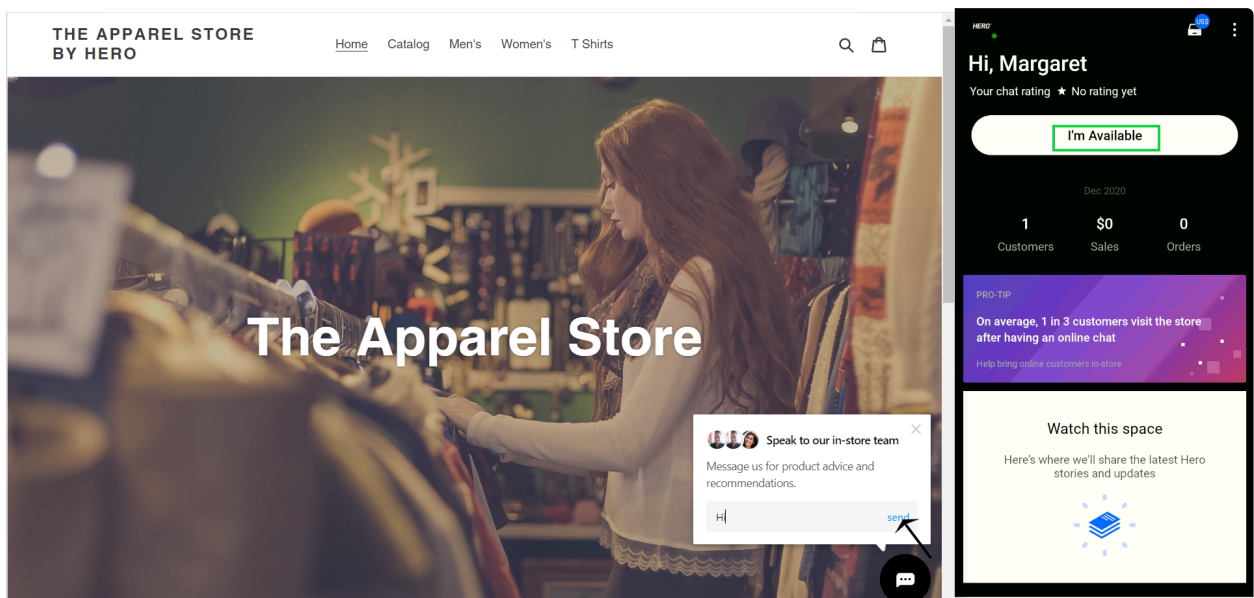
Name	Status	Type	Initiator	Size	Time	Waterfall
loader.js	200	script	(index):933	23.8 kB	546 ms	
attribution	204	Other		0 B	226 ms	
display?appid=0efc7ab2-64...	200	xhr	shop_events.js...	622 B	277 ms	
attribution	200	xhr	shop_events.js...	888 B	165 ms	
plugin.5.23.0.js	200	script	loader.js:1	56.4 kB	196 ms	
chunk.519.9c7ffa6d00b754a...	200	script	plugin.5.23.0.js...	134 kB	218 ms	
shopper?appid=0efc7ab2-6...	200	xhr	chunk.519.9c7...	8.0 kB	107 ms	
settings?appid=0efc7ab2-6...	200	xhr	chunk.519.9c7...	1.7 kB	91 ms	
metrics	204	Other		0 B	87 ms	
me	200	text/pl...	Other	0 B	101 ms	
metrics	200	xhr	chunk.519.9c7...	456 B	62 ms	
me	200	xhr	chunk.519.9c7...	718 B	71 ms	
ping	200	text/pl...	Other	0 B	71 ms	
chunk.module-twilio-chat.5...	200	script	plugin.5.23.0.js...	74.5 kB	167 ms	
1607d8b40d073354d54b6c...	200	png	(index)	1.5 kB	49 ms	
metrics	204	Other		0 B	93 ms	
ping	200	xhr	chunk.519.9c7...	588 B	109 ms	
lineup?appid=0efc7ab2-64...	200	xhr	chunk.519.9c7...	879 B	116 ms	
metrics	200	xhr	chunk.519.9c7...	457 B	80 ms	
metrics	204	Other		0 B	102 ms	
b8966a9b628532bd609c55...	200	png	(index)	906 B	55 ms	
metrics	200	xhr	chunk.519.9c7...	457 B	71 ms	

2. Make yourself available on the Store App.

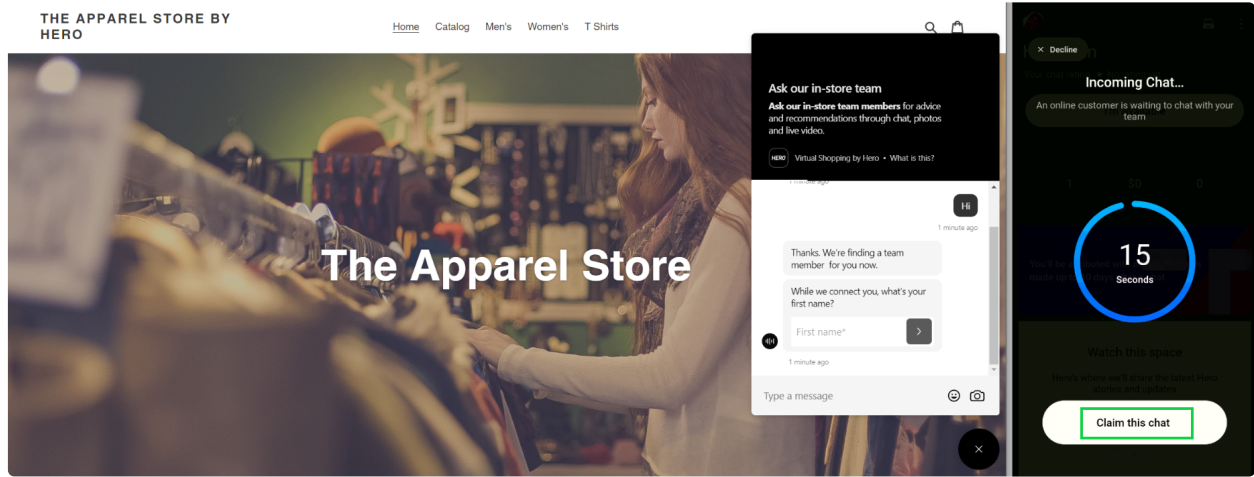


Steps to test

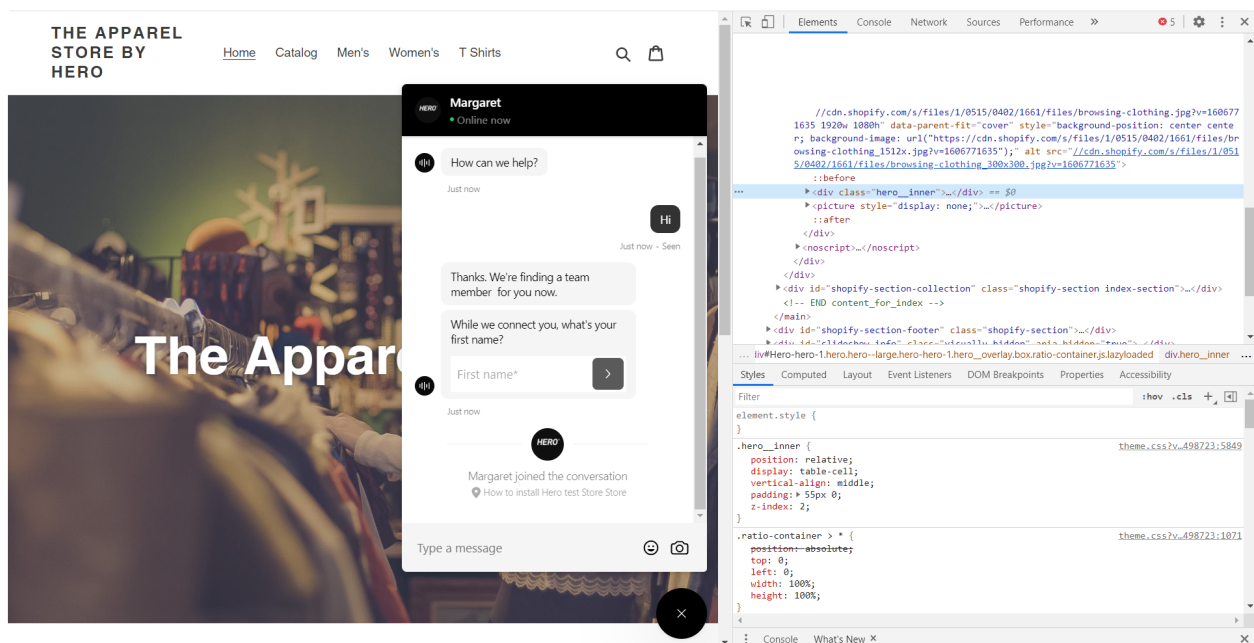
1. Start a Virtual Shopping chat.



2 Accept chat on the Store App.



3. Right-click on your website page and click on inspect to display developer tools.



4. Open the Network tab to display the Network panel. Ensure you have checked the 'Disable cache' box.

THE APPAREL STORE BY HERO
Home Catalog Men's Women's T Shirts

HERO
I'm live in-store at How to install Hero

Virtual Shopping by Hero • What is this?

While we connect you, what's your first name?

First name*

4 minutes ago

HERO
Margaret joined the conversation
How to install Hero test Store Store

HERO
I'm an expert at How to install Hero
Sent from Mobile - 4 minutes ago

Type a message

Network Console:
Filter: tracking
8 requests | 542 kB transferred | 537 kB resources

Name	Status	Type	Initiator	Size	Time	Waterfall
browsing-clothing_1512xjp...	200	webp	lazysizes.js?v=...	257 kB	537 ms	
red-plaid-shirt_925x_b9b01...	200	webp	lazysizes.js?v=...	4.4 kB	535 ms	
womens-red-t-shirt_925x_b...	200	webp	lazysizes.js?v=...	3.7 kB	554 ms	
smiling-woman-poses_925x...	200	webp	lazysizes.js?v=...	8.3 kB	550 ms	
young-man-in-bright-fashi...	200	webp	lazysizes.js?v=...	10.0 kB	550 ms	
browsing-clothing_1512xjp...	200	webp	(index)	257 kB	221 ms	
ping	200	xhr	chunk519.9c7...	587 B	101 ms	
ping	200	text/pl...	Other	0 B	243 ms	

6. Type in the filter text box: **tracking**

THE APPAREL STORE BY HERO
Home Catalog Men's Women's T Shirts

HERO
I'm live in-store at How to install Hero

Virtual Shopping by Hero • What is this?

While we connect you, what's your first name?

First name*

10 minutes ago

HERO
Margaret joined the conversation
How to install Hero test Store Store

HERO
I'm an expert at How to install Hero
Sent from Mobile - 10 minutes ago

Type a message

Network Console:
Filter: tracking
0 / 22 requests | 0 B / 547 kB transferred | 0 B / 538 kB resources

Name	Status	Type	Initiator	Size	Time	Waterfall
------	--------	------	-----------	------	------	-----------

7. On website carry out a shopper action that would trigger the Shopper Event i.e. View a Product page to fire the 'Product view' event.

THE APPAREL STORE BY HERO

Home Catalog Men's Women's T Shirts

Chequered Red Shirt

\$50.00

Title: Chequered Red Shirt

Color: Red

[ADD TO CART](#)

[BUY IT NOW](#)

Classic mens plaid flannel shirt with long sleeves, in chequered style, with two chest pockets.

[SHARE](#) [TWEET](#) [PIN IT](#)

Network Log:

Name	Status	Type	Initiator	Size	Time	Waterfall
consent-tracking-ap.js	200	script	load_feature...	2.1 kB	57 ms	
tracking	200	xhr	shop_events.li...	548 B	159 ms	
tracking	204	Other	Other	0 B	92 ms	

3 / 80 requests | 2.6 kB / 739 kB transferred | 4.2 kB / 2.7 MB resources | Finish: 6.20 s | DOMContentLoaded: 1.25 s | Load: 2.0 s

8. As an option to further filter the Network log, you can click on the XHR tab to only show the Shopper Event tracking resource with a 200 HTTP response code

THE APPAREL STORE BY HERO

Home Catalog Men's Women's T Shirts

Chequered Red Shirt

\$50.00

Title: Chequered Red Shirt

Color: Red

[ADD TO CART](#)

[BUY IT NOW](#)

Classic mens plaid flannel shirt with long sleeves, in chequered style, with two chest pockets.

[SHARE](#) [TWEET](#) [PIN IT](#)

Network Log:

Name	Status	Type	Initiator	Size	Time	Waterfall
tracking	200	xhr	shop_event...	547 B	127 ...	

1 / 91 requests | 547 B / 1.5 MB transferred | 15 B / 6.1 MB resources | Finish: 11.42 s

9. Click on the tracking request in the Network log that displays 200 as the HTTP response code. Click on the Headers tab to display the request payload

11. Expand the request payload to view the Shopper Event

Results

Correct result

- ✅ The Shopper Event should be displayed in the request payload



Chequered Red Shirt

\$50.00

Title
Chequered Red Shirt

Color
Red

ADD TO CART

BUY IT NOW

```
Request URL: https://api.usehero.com/webplugin/tracking
Request Method: POST
Status Code: 200
Remote Address: 54.74.57.107:443
Referrer Policy: strict-origin-when-cross-origin

Response Headers (18)
Request Headers (18)
Request Payload
[applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c",...]
applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c"
events: [{"id": "73d13f6f-5ca8-4dd0-a1de-34ab2bfc892c", date: "2020-12-16T15:10:31.037Z",...}]
0: {"id": "73d13f6f-5ca8-4dd0-a1de-34ab2bfc892c", date: "2020-12-16T15:10:31.037Z",...}
id: "73d13f6f-5ca8-4dd0-a1de-34ab2bfc892c"
products: [{"id": "1606771598", name: "Chequered Red Shirt",...}]
0: {"id": "1606771598", name: "Chequered Red Shirt",...}
brand: "Hero"
category: "Shirt"
currency: "USD"
id: "1606771598"
image: "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/red-plaid-shirt_925x_b9b019f9-01a5-4515-89b0-ae6da80f0671_925x.jpg?v=1606771598"
location: "https://the-apparel-store-by-hero.myshopify.com/products/chequered-red-shirt"
name: "Chequered Red Shirt"
price: 50
variant: "Chequered Red Shirt / Red"
type: "ecommerce:detail"
```

✓ The Shopper Event should show on the Store App



Chequered Red Shirt

\$50.00

Title
Chequered Red Shirt

Color
Red

ADD TO CART

BUY IT NOW

```
Request URL: https://api.usehero.com/webplugin/tracking
Request Method: POST
Status Code: 200
Remote Address: 54.74.57.107:443
Referrer Policy: strict-origin-when-cross-origin

Response Headers (18)
Request Headers (18)
Request Payload
[applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c",...]
applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c"
events: [{"id": "73d13f6f-5ca8-4dd0-a1de-34ab2bfc892c", date: "2020-12-16T15:10:31.037Z",...}]
0: {"id": "73d13f6f-5ca8-4dd0-a1de-34ab2bfc892c", date: "2020-12-16T15:10:31.037Z",...}
id: "73d13f6f-5ca8-4dd0-a1de-34ab2bfc892c"
products: [{"id": "1606771598", name: "Chequered Red Shirt",...}]
0: {"id": "1606771598", name: "Chequered Red Shirt",...}
brand: "Hero"
category: "Shirt"
currency: "USD"
id: "1606771598"
image: "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/red-plaid-shirt_925x_b9b019f9-01a5-4515-89b0-ae6da80f0671_925x.jpg?v=1606771598"
location: "https://the-apparel-store-by-hero.myshopify.com/products/chequered-red-shirt"
name: "Chequered Red Shirt"
price: 50
variant: "Chequered Red Shirt / Red"
type: "ecommerce:detail"
```

Shopper 1288797
Online

Viewed item
US\$50.00
Chequered Red Shirt

34 seconds ago

Shopper 1288797 started a chat
Just now

Hi

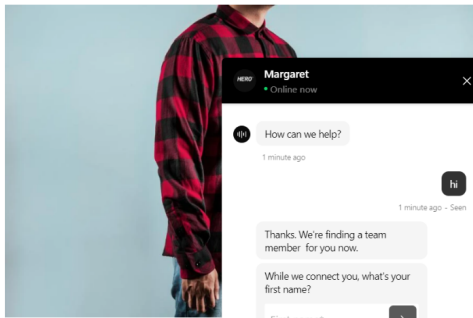
You accepted this chat

Hi, I'm Margaret | I'm an expert at How to Install Hero

Type a message...

Incorrect result

- ✗ Shopper Event is not visible in the Network tab
- ✗ The Shopper Event is not captured on the Store App



Chequered Red Shir

\$50.00

Title

Chequered Red Shirt

Color

Margaret
• Online now

Hi

How can we help?

1 minute ago

hi

1 minute ago - Seen

Thanks. We're finding a team member for you now.

While we connect you, what's your first name?

First name*

1 minute ago

Margaret joined the conversation

How to install Hero test Store Store

Type a message

Elements Console Network Sources Performance Memory Application

tracking

Blocked Requests

Name	Status	Type	Initiator	Size	Time	Waterfall
consent-tracking-api.js	200	script	load feature-24ff...	2.2 kB	83 ms	

1 / 112 requests | 2.2 kB / 1.3 MB transferred | 4.2 kB / 3.5 MB resources | Finish: 1.2 min | DOMContentLoaded: 1.4

Shopper 1288988
• Online

No items viewed

Shopper 1288988 started a chat
3 minutes ago

hi

You accepted this chat

Hi, I'm Margaret | I'm an expert at How to install

Type a message...

Troubleshooting

Shopper Event tracking

There are many potential reasons why your implemented Shopper Events may not be behaving as expected. To help you solve the problem quickly, here are common challenges you may come across when setting up Shopper Events with simple solutions.



We recommend you begin troubleshooting with [All Shopper Events](#) before reviewing articles on specific Shopper Events.



All Shopper Events



Product View



Purchase



Basket Add



Basket Remove



Category View



Search

All Shopper Events

Shopper Event Troubleshooting

Here are a series of common scenarios you may face when testing the set up of any Shopper Event.

✓ We recommend starting any troubleshooting here before reviewing articles on specific Shopper Events.

Shopper Event is deployed but does not show in the Network tab

If you have deployed the Tracking API but are not seeing the tracking request in the Network log, then this means that the Shopper Event may not have been set up correctly or further steps are required to see the tracking endpoint in the Network tab.

Reasons why this may be happening:

1. A Virtual Shopping chat has not been initiated

You will need to start an On-site Chat to see the tracking endpoint in the Network tab. To do this type and send an On-site Chat message - as a shopper - and accept the incoming chat on the Store App.

2. Tracking API has not been deployed to the required pages

The tracking API needs to be deployed to the necessary pages for the Shopper Event to fire on these pages:

- Product view: all Product detail pages (PDPs)
- Purchase: the checkout confirmation page
- Basket add: Any page where the shopper could add an item to a basket
- Basket remove: Any page where the shopper can remove an item from the basket
- Category view: Any page with a category structure
- Search: Any page where the shopper could search for an item

3. The event deployed does not follow the correct structure

Ensure that the Shopper Event tracking tag is deployed with all the required fields for the event - see above for guidance for each event.

Shopper Event fires on Network tab but does not show on the Store app

Reasons why this may be happening:

Incorrect event type value passed into the event.

Sometimes it's a just small typo for example, `ecommerce: detail` instead of `ecommerce:detail`. To ensure that the event type value is added correctly, check [here](#).

✗ Example of incorrect set up of 'type' field value:

The screenshot shows the Network tab in a browser's developer tools. A request named 'tracking' is selected. The 'Request Payload' is expanded, showing a JSON object with a 'products' array. The first object in the array has a 'type' field set to 'ecommerce: detail1', which is highlighted with a green box. To the right, a chat window for 'Shopper 1282796' is visible, showing a message 'No items viewed' in a dark box.

Example correct setup of 'type' field value

The screenshot shows the Network tab in a browser's developer tools. A request named 'tracking' is selected. The 'Request Payload' is expanded, showing a JSON object with a 'products' array. The first object in the array has a 'type' field set to 'ecommerce:detail1', which is highlighted with a green box. To the right, a chat window for 'Shopper 1282796' is visible, showing a message 'Viewed item' with a red sports tee product image.

Products object not passed as an array of objects

This would prevent the Shopper Event from showing as expected on the Store App.

The screenshot shows a network request in the Chrome DevTools console. The request is a POST to a tracking endpoint. The payload includes a 'products' array with one object:

```

{
  "products": [
    {
      "id": "1606771604",
      "name": "Red Sports Tee",
      "brand": "Hero",
      "category": "Top",
      "currency": "USD",
      "id": "1606771604",
      "image": "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/womens-red-t-shirt_925x_b78d289c-e4d9-4b00-bde3-671c43b03841_925x.jpg?v=1606771604",
      "location": "https://the-apparel-store-by-hero.myshopify.com/products/red-sports-tee",
      "name": "Red Sports Tee",
      "price": 50,
      "variant": "Red Sports Tee / Red"
    }
  ]
}

```

The response shows a chat interface with a 'No items viewed' message, indicating that the purchase event was not processed correctly.

Purchase subfields are passed as an array instead of an object

This would prevent the Purchase Shopper Event from showing as expected on the Store app.

The screenshot shows a network request in the Chrome DevTools console. The request is a POST to a tracking endpoint. The payload includes a 'purchase' array with one object:

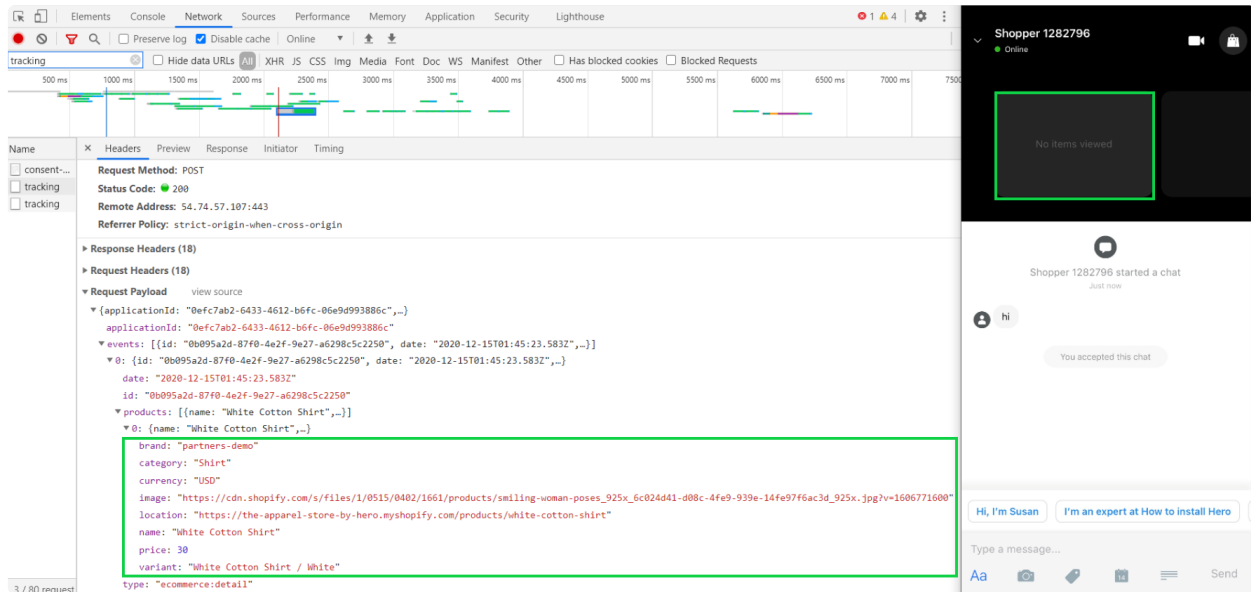
```

{
  "purchase": [
    {
      "currency": "USD",
      "id": "1013",
      "total": 63.54,
      "shippingCost": 13.54,
      "currency": "USD",
      "id": "1013",
      "shippingCost": 13.54,
      "shippingMethod": "First Class Package International",
      "subtotal": 50,
      "tax": 0,
      "total": 63.54
    }
  ]
}

```

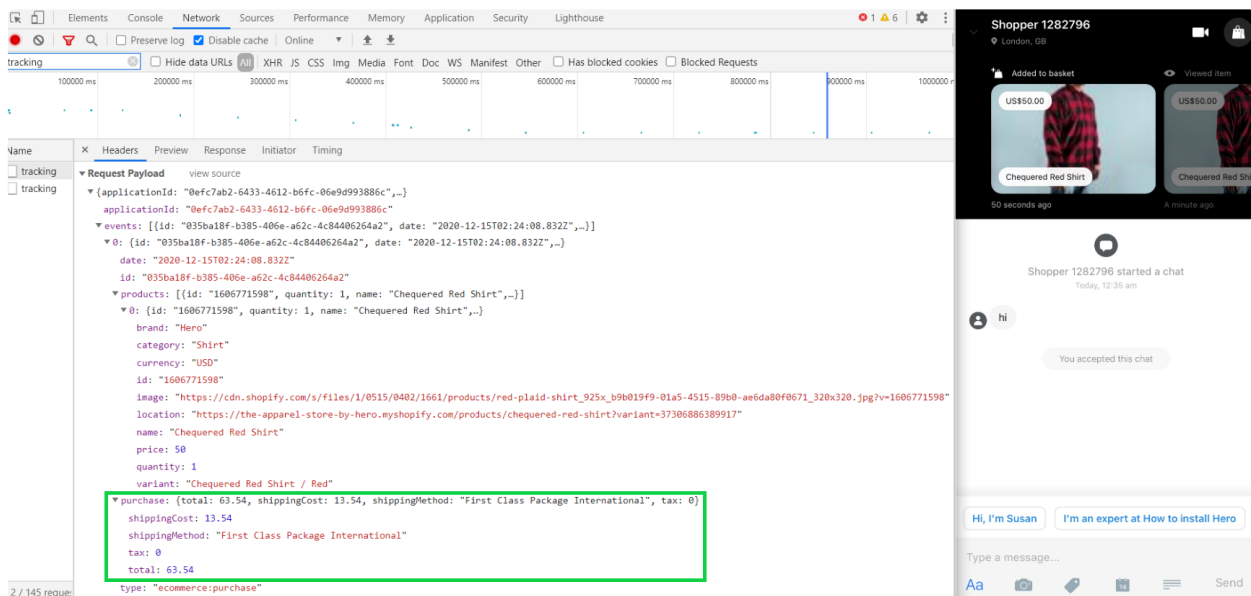
The response shows a chat interface with a 'No items viewed' message, indicating that the purchase event was not processed correctly.

Product ID or SKU value is missing within the product array



Subtotal, Order ID and currency fields are not provided in the Purchase event causing the Purchase Event to not show on the Store app

The example below shows the last Shopper Event captured on the Store app - a Basket Add event. The Purchase Shopper Event failed to show on the app due to an error during set up of the Purchase event (missing the subtotal, currency and ID subfields).



Shopper Event not showing as expected on the Store app

There can be instances, where the Shopper Event is displayed in the Network tab, but not captured as expected on the app. If this is the case, then it would mean that the data being passed into the event is incorrect, or does not meet the implementation requirements outlined [here](#).

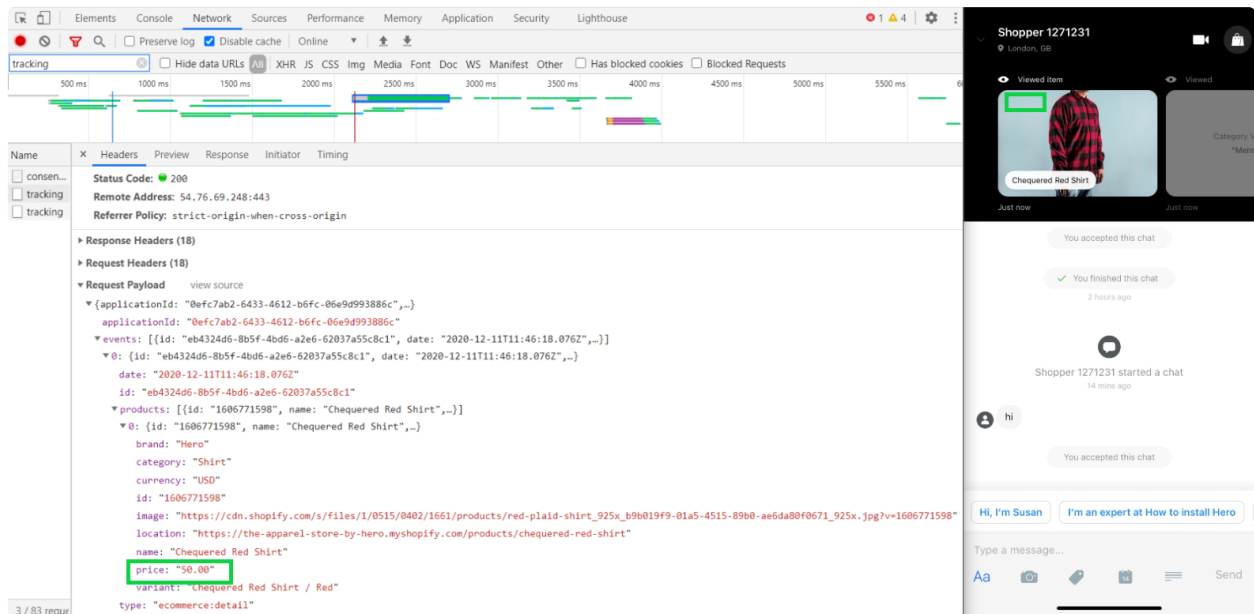
Read on for common tracking implementation issues specific to each Shopper Event:

Product View

Shopper Event Troubleshooting

Product price not displaying on the Product View Event card

This issue would occur as a result of the price value being passed as a string instead of a number. It is important to ensure that this field value is provided as a number so that it shows as expected on the Store App.

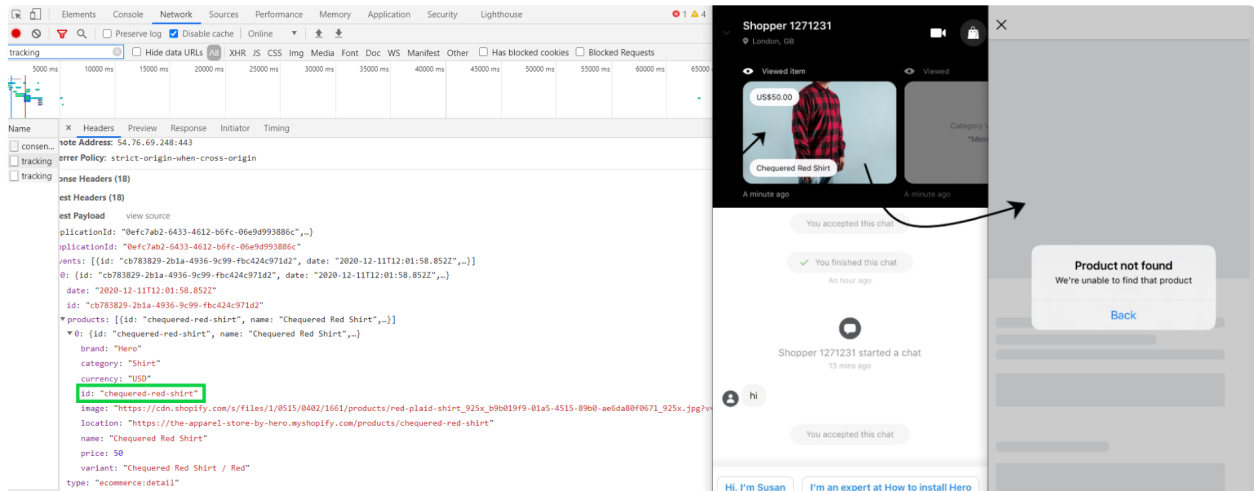


'Product not found' message shows on the Store app when a user taps on a Product View Event card

If your Product Feed has not yet been indexed then the 'Product not found' message will display when you tap on the Shopper Event card.

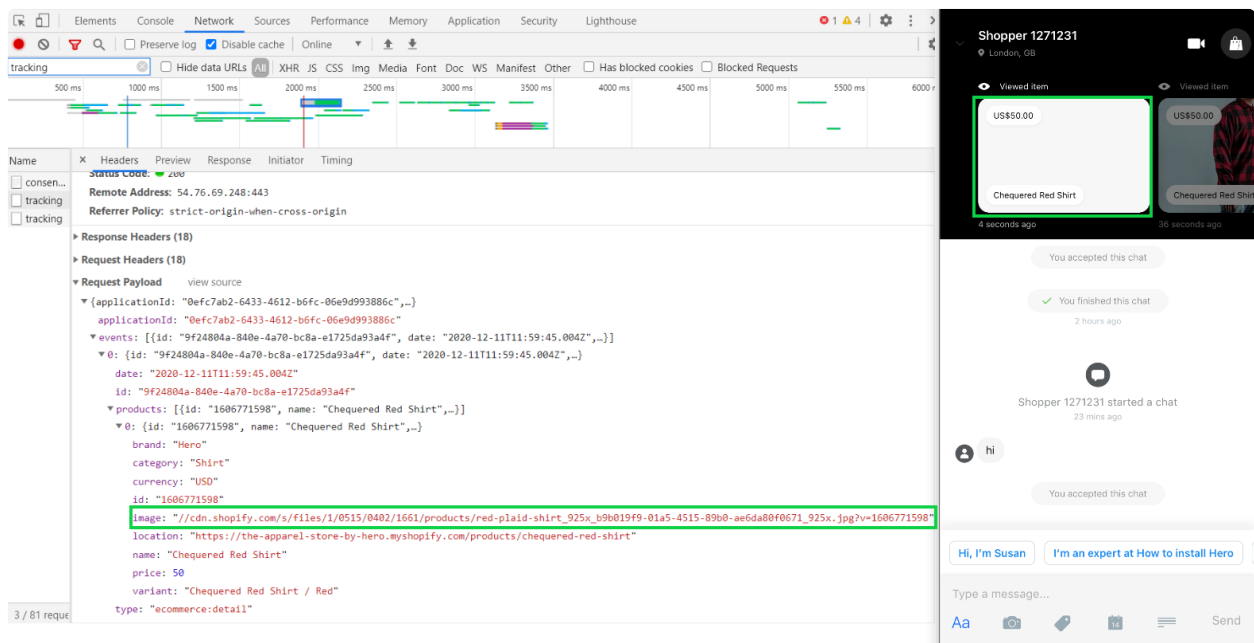
If your Product Feed has been indexed successfully - products are displaying in the product catalogue - then you should be able to tap on the Shopper Event card to view the product detail page on the Store app. If you are seeing the 'Product not found' message, this would mean that the product ID provided in the event, does not match the product ID provided in the Product Feed.

For example, if the product ID / SKU provided in the feed is `1606771598` and the ID provided in the event does not match this exact value, then this will result in the 'Product not found' error message.



Product image missing from the Product View Event card

This issue occurs if the Image URL provided for the Shopper Event is either missing from the product array or is not provided as https or is not a valid URL. All three scenarios can result in the product image to not display on the product card for the Shopper Event.



Location URL is missing / is not provided as https

THE APPAREL STORE BY HERO

Chequered Red Shirt
\$50.00

Title: Chequered Red Shirt

Color: Red

ADD TO CART

BUY IT NOW

Network Request Details:

- Request Method: POST
- Status Code: 200
- Remote Address: 54.76.69.248:443
- Referrer Policy: strict-origin-when-cross-origin
- Request Payload:


```
{
    "applicationId": "0efc7ab2-6433-4612-b6fc-06e9d993886c",
    "applicationId": "0efc7ab2-6433-4612-b6fc-06e9d993886c",
    "events": [
      {
        "id": "7bca18c7-38d9-4346-91cf-d14e926424a2",
        "date": "2020-12-17T12:20:52.7062Z"
      },
      {
        "id": "7bca18c7-38d9-4346-91cf-d14e926424a2",
        "date": "2020-12-17T12:20:52.7062Z"
      }
    ],
    "products": [
      {
        "id": "1606771598",
        "name": "Chequered Red Shirt",
        "brand": "Hero",
        "category": "Shirt",
        "currency": "USD",
        "id": "1606771598",
        "image": "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/red-plaid-shirt_925x_b9b019f9-01a5-4515-89b0-ae6da80f0671_925x.jpg?v=1606771598",
        "location": "/the-apparel-store-by-hero.myshopify.com/products/chequered-red-shirt",
        "name": "Chequered Red Shirt",
        "price": 50,
        "variant": "Chequered Red Shirt / Red",
        "type": "ecommerce:detail"
      }
    ]
  }
```

Incorrect currency value provided in the currency field

This should be the actual currency of the product as seen on your web pages and Product Feed.

THE APPAREL STORE BY HERO

Chequered Red Shirt
\$50.00

Title: Chequered Red Shirt

Color: Red

ADD TO CART

BUY IT NOW

Network Request Details:

- Request Method: POST
- Status Code: 200
- Remote Address: 54.74.57.107:443
- Referrer Policy: strict-origin-when-cross-origin
- Request Payload:

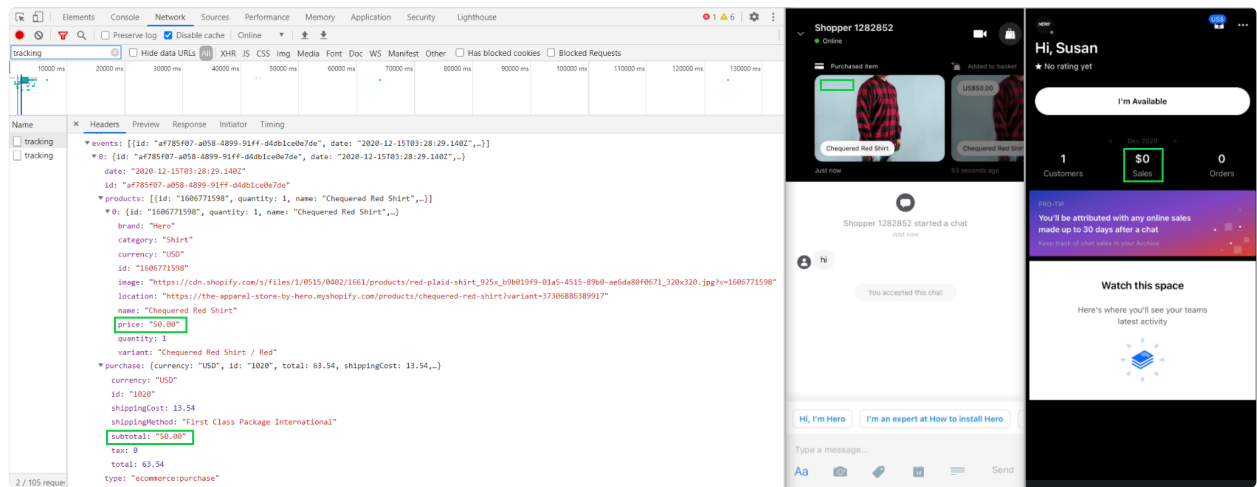

```
{
    "applicationId": "0efc7ab2-6433-4612-b6fc-06e9d993886c",
    "applicationId": "0efc7ab2-6433-4612-b6fc-06e9d993886c",
    "events": [
      {
        "id": "b6c32b15-2253-42b0-a7c8-b1bf227edbf6",
        "date": "2020-12-17T12:26:17.9632Z"
      },
      {
        "id": "b6c32b15-2253-42b0-a7c8-b1bf227edbf6",
        "date": "2020-12-17T12:26:17.9632Z"
      }
    ],
    "products": [
      {
        "id": "1606771598",
        "name": "Chequered Red Shirt",
        "brand": "Hero",
        "category": "Shirt",
        "currency": "EUR",
        "id": "1606771598",
        "image": "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/red-plaid-shirt_925x_b9b019f9-01a5-4515-89b0-ae6da80f0671_925x.jpg?v=1606771598",
        "location": "https://the-apparel-store-by-hero.myshopify.com/products/chequered-red-shirt",
        "name": "Chequered Red Shirt",
        "price": 50,
        "variant": "Chequered Red Shirt / Red",
        "type": "ecommerce:detail"
      }
    ]
  }
```

Purchase

Shopper Event Troubleshooting

Product Price missing from Purchase Event card and the purchase sale is not captured on the home screen

This issue is as a result of the subtotal (of the purchase subfield) and price (of the subfield in product array) provided as a string in the 'Purchase' Shopper Event. Both values should be sent through the event as a number to successfully capture the sale on the Store App.

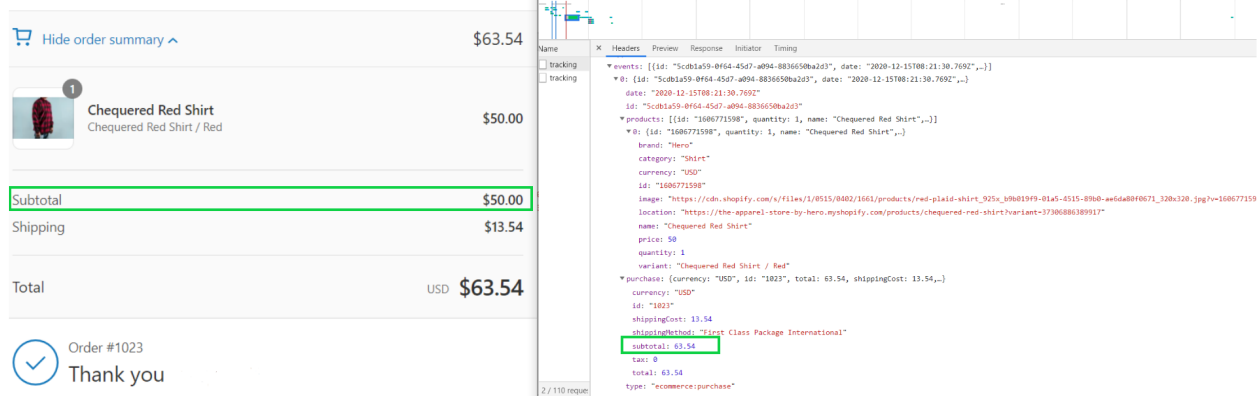


Incorrect subtotal value shown on the Store App home screen and Dashboard

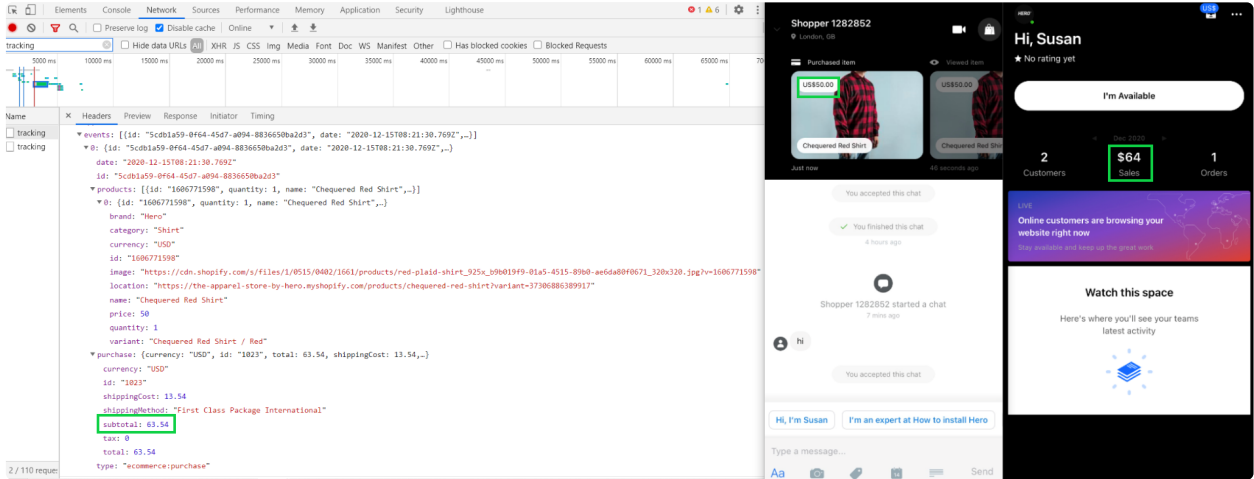
This can occur when the subtotal value passed in the 'Purchase' Shopper Event does not match the actual cost of the product, excluding the shipping costs.

To accurately capture the sale attributed to the store team member, it is important to ensure that the subtotal value does not include the shipping cost of the sale.

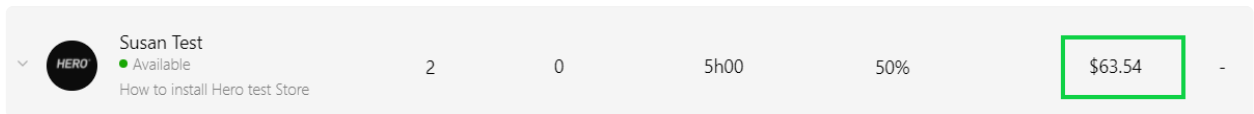
The Apparel Store by Hero



Incorrect subtotal value is provided in 'Purchase' Shopper Event.



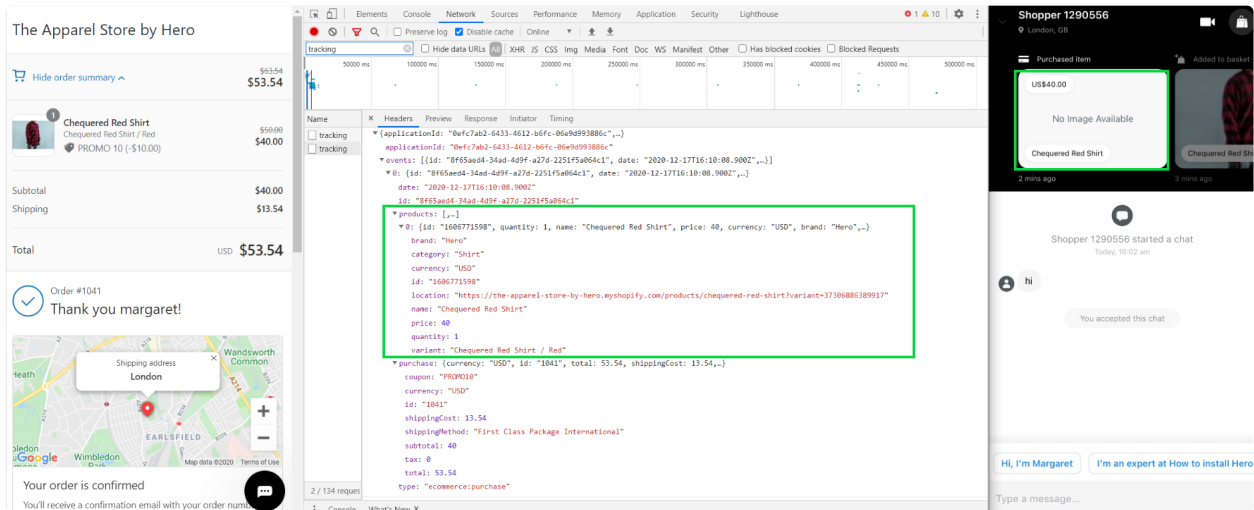
Incorrect subtotal value - including shipping cost - is attributed to the Hero App home screen.



Dashboard shows incorrect subtotal value provided in the Purchase Shopper Event.

Product image missing from the Purchase Event card

This issue occurs if the Image URL provided for the Shopper Event is either missing from the product array or is not provided as https or is not a valid URL. All three scenarios can result in the product image to not display on the shopper event product card.



No image URL in the Purchase Shopper Event means no image in the Shopper Event card.

Product name missing from Purchase Event card

If the name subfield passed into the Shopper Event is empty, then this would result in no product name displaying on the Shopper Event card.

The Apparel Store by Hero

Order #1045
Thank you margaret!

Shipping address: London

Your order is confirmed

Shopper 1290556 started a chat
Today, 10:02 am

hi

You accepted this chat

Hi, I'm Margaret | I'm an expert at How to install Hero

Type a message... Send

```

Name: tracking
Headers: applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c"
Response:
  events: [{"id": "01e8451d-7714-422f-85c8-e67865c61190", date: "2020-12-17T16:56:58.874Z", ...}]
  products: [{"id": "160671598", quantity: 1, name: "", price: 40, ...}]
  purchase: {currency: "USD", id: "1045", total: 53.54, shippingCost: 13.54, ...}
  location: "https://the-apparel-store-by-hero.myshopify.com/products/chequered-red-shirt?variant=37206886389917"
  name: ""
  price: 40
  quantity: 1
  variant: "Chequered Red Shirt / Red"
  coupon: "PROMO10"
  currency: "USD"
  id: "1045"
  shippingCost: 13.54
  shippingMethod: "First Class Package International"
  subtotal: 40
  tax: 0
  total: 53.54
  type: "ecommerce.purchase"

```

Location URL is missing / is not provided as https

The Apparel Store by Hero

Order #1042
Thank you margaret!

Shipping address: London

Your order is confirmed

Shopper 1290556 started a chat
Today, 10:02 am

hi

You accepted this chat

Hi, I'm Margaret | I'm an expert at How to install Hero

Type a message... Send

```

Name: tracking
Headers: applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c"
Response:
  events: [{"id": "e7b92028-96c0-48b1-ba48-6b82712e32ea", date: "2020-12-17T16:43:00.536Z", ...}]
  products: [{"id": "160671598", quantity: 1, name: "Chequered Red Shirt", price: 40, ...}]
  purchase: {currency: "USD", id: "1042", total: 53.54, shippingCost: 13.54, ...}
  location: "/the-apparel-store-by-hero.myshopify.com/products/chequered-red-shirt?variant=37306886389917"
  name: "Chequered Red Shirt"
  price: 40
  quantity: 1
  variant: "Chequered Red Shirt / Red"
  coupon: "PROMO10"
  currency: "USD"
  id: "1042"
  shippingCost: 13.54
  shippingMethod: "First Class Package International"
  subtotal: 40
  tax: 0
  total: 53.54
  type: "ecommerce.purchase"

```

currency field has incorrect value

This should be the 3-letter ISO 4217 code for the currency of the transaction (e.g. GBP for pound sterling).

The Apparel Store by Hero

Hide order summary \$62.54
\$53.54

Chequered Red Shirt
Chequered Red Shirt / Red
PROMO 10 (-\$10.00) **\$40.00**

Subtotal **\$40.00**
Shipping **\$13.54**

Total USD **\$53.54**

Order #1048
Thank you margaret!

Shipping address
London

Your order is confirmed
You'll receive a confirmation email with...

```

applicationId: "0efc7ab2-6433-4612-b6fc-06e94993886c"
events: [{"id": "f5f3e66d-7fc8-4636-9973-35fab9ab33c3", date: "2020-12-17T17:49:58.333Z", ...}, {"id": "f5f3e66d-7fc8-4636-9973-35fab9ab33c3", date: "2020-12-17T17:49:58.333Z", ...}],
products: [{"id": "1606771598", quantity: 1, name: "Chequered Red Shirt", price: 40, ...}],
purchase: {currency: "USD", id: "1048", total: 53.54, shippingCost: 13.54, ...},
coupon: "PROMO10",
currency: "USD",
id: "1048",
shippingCost: 13.54,
shippingMethod: "First Class Package International",
subtotal: 40,
tax: 0,
total: 53.54,
type: "ecommerce:purchase"

```

quantity, shippingCost, subtotal, tax and total value provided as strings instead of as numbers

These values including price and basketQuantity values should be provided in the correct format of a number.

The Apparel Store by Hero

Hide order summary \$63.54
\$53.54

Chequered Red Shirt
Chequered Red Shirt / Red
PROMO 10 (-\$10.00) **\$40.00**

Subtotal **\$40.00**
Shipping **\$13.54**

Total USD **\$53.54**

Order #1054
Thank you margaret!

Shipping address
London

Your order is confirmed

```

date: "2020-12-18T03:20:08.016Z"
id: "9bf49281-43c9-447f-8f32-25f95e4f414b"
products: [{"id": "1606771598", quantity: 1, name: "Chequered Red Shirt", ...}],
purchase: {currency: "USD", id: "1054", total: "53.54", shippingCost: "13.54", ...},
coupon: "PROMO10",
currency: "USD",
id: "1054",
shippingCost: "13.54",
shippingMethod: "First Class Package International",
subtotal: "40.00",
tax: "0.00",
total: "53.54",
type: "ecommerce:purchase"

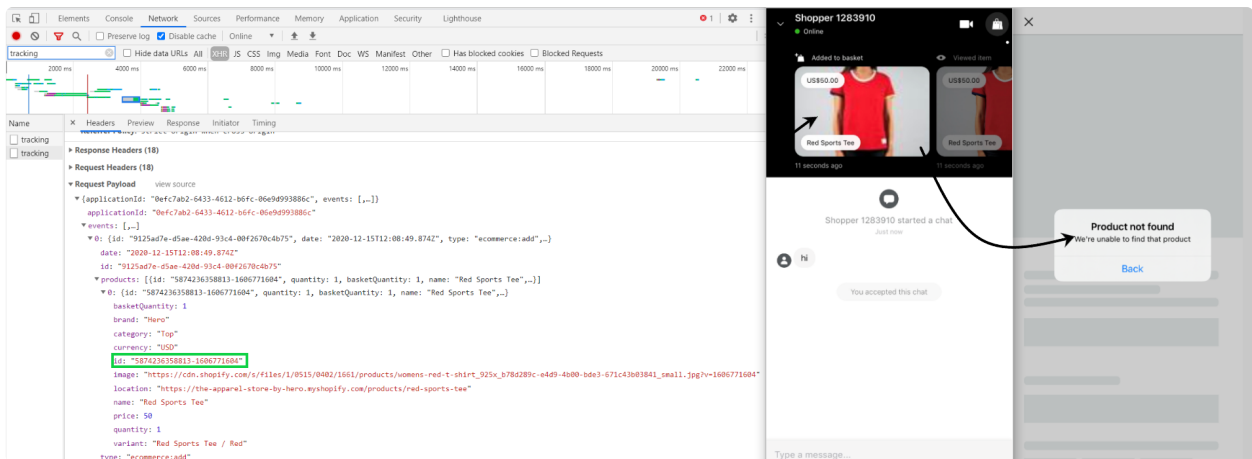
```


Basket Add

Shopper Event Troubleshooting

'Product not found' message appearing on the Store App when a user taps on a Basket Add Event card

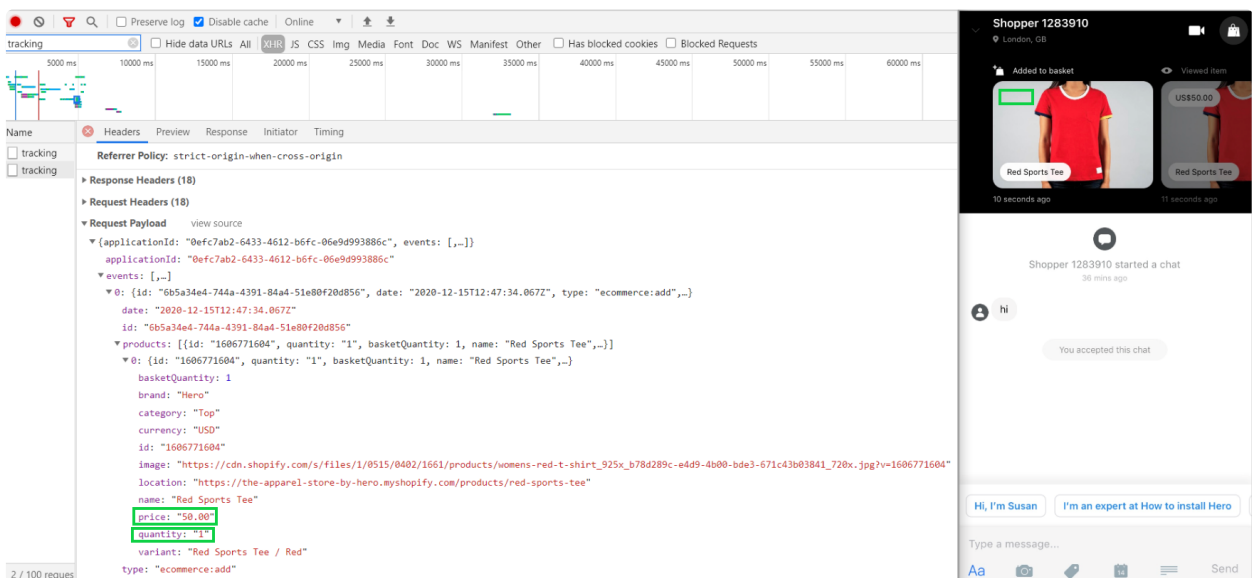
Assuming that the Product Feed has been indexed and products are displaying in the product catalogue, if this issue is present, then this would mean that the product ID (SKU or variant ID) provided in the Shopper Event, does not match the product ID provided in the Product Feed for that specific product.



Here the product ID provided in the event does not match the ID / SKU in the Product Feed.

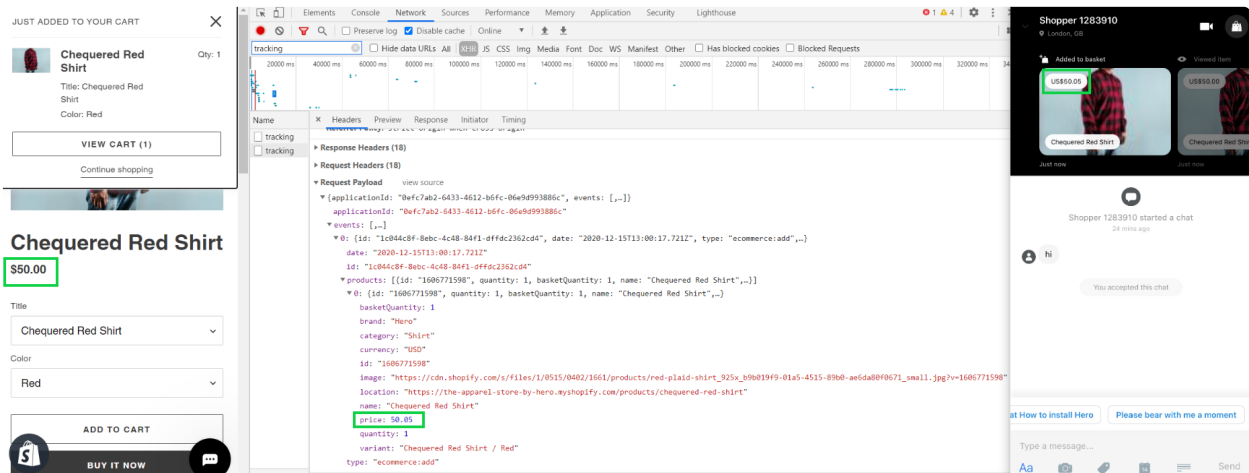
Product price value missing from Basket Add Event card

If the price value is missing from the Shopper event card, this would mean that the price value passed in the 'Add to basket' Shopper Event has been provided as a string and not a number. It is important to ensure that price, quantity and basketQuantity values are implemented as a number and not a string in the Shopper Event.



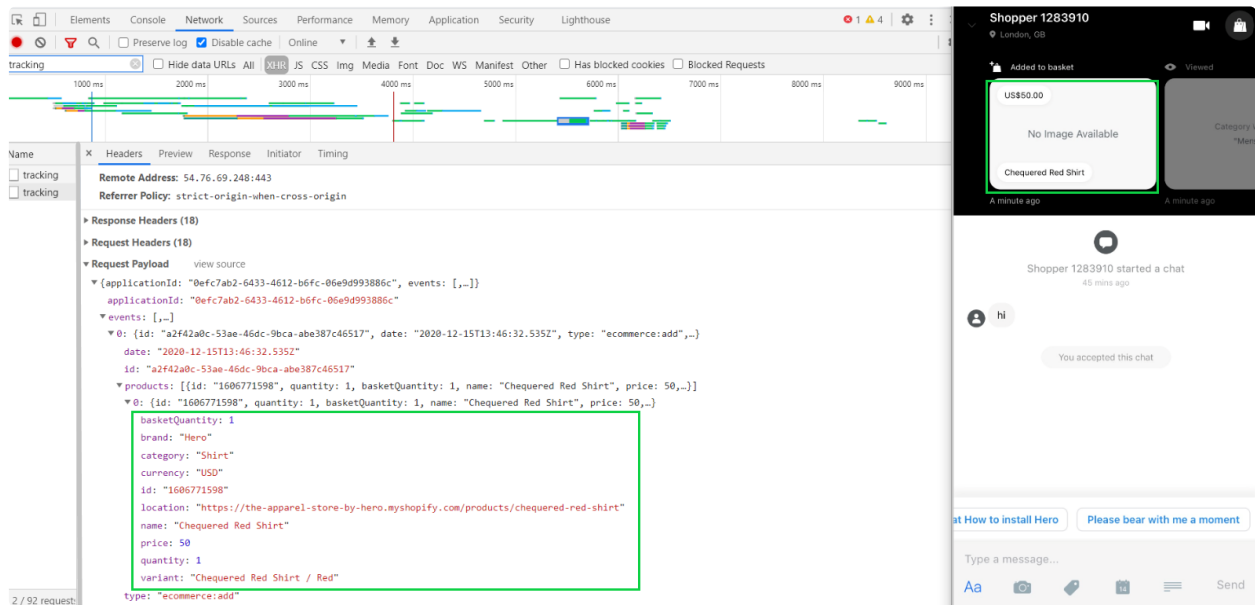
Price on Basket Add Event card does not match the current price of the product

This issue occurs when the price provided within the product array of the Shopper Event, does not match the current price of the product. It is important to ensure that the product price within the event matches the current price of the product provided on your website and also in your Product Feed.



Product image missing from Basket Add Event card

This issue displayed on the app would occur if the image field / the image URL value is missing within the product array or if the URL provided is not https and / or is not a valid URL. All three scenarios would result in the product image to not display as expected on the app for the shopper event card.



Here, the image URL is missing from the Shopper Event.

Location URL is missing / is not provided as https

JUST ADDED TO YOUR CART

Red Sports Tee Qty: 1

Title: Red Sports Tee
Color: Red

VIEW CART (1)

Continue shopping

Red Sports Tee
\$50.00

Title: Red Sports Tee

Color: Red

ADD TO CART

BUY IT NOW

```

Name: tracking
Headers:
  sec-fetch-dest: empty
  sec-fetch-mode: cors
  sec-fetch-site: cross-site
  user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Request Payload:
  applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c"
  applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c"
  events: [...]
  0: {id: "d500351e-83e6-4e3b-b33d-4095c6c30b82", date: "2020-12-22T14:11:09.851Z", type: "ecommerce:add",...}
  date: "2020-12-22T14:11:09.851Z"
  id: "d500351e-83e6-4e3b-b33d-4095c6c30b82"
  products: [(id: "1606771604", quantity: 1, basketQuantity: 1, name: "Red Sports Tee",...)]
  0: {id: "1606771604", quantity: 1, basketQuantity: 1, name: "Red Sports Tee",...}
    basketQuantity: 1
    brand: "Hero"
    category: "Top"
    currency: "USD"
    id: "1606771604"
    image: "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/womens-red-t-shirt_925x_b78d289c-e4d9-4b00-bde3-671c43b03841_small.jpg?v=1606771604"
    location: "/the-apparel-store-by-hero.myshopify.com/products/red-sports-tee"
    name: "Red Sports Tee"
    price: 50
    quantity: 1
    variant: "Red Sports Tee / Red"
    type: "ecommerce:add"
  
```

Currency field or currency value is incorrect or missing within the product array

JUST ADDED TO YOUR CART

Red Sports Tee Qty: 1

Title: Red Sports Tee
Color: Red

VIEW CART (1)

Continue shopping

Red Sports Tee
\$50.00

Title: Red Sports Tee

Color: Red

ADD TO CART

BUY IT NOW


```

Name: tracking
Headers:
  sec-fetch-dest: empty
  sec-fetch-mode: cors
  sec-fetch-site: cross-site
  user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
Request Payload:
  applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c"
  applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c"
  events: [...]
  0: {id: "dd09eada-394a-42bc-8558-b4a51032a425", date: "2020-12-22T14:21:15.224Z", type: "ecommerce:add",...}
  date: "2020-12-22T14:21:15.224Z"
  id: "dd09eada-394a-42bc-8558-b4a51032a425"
  products: [(id: "1606771604", quantity: 1, basketQuantity: 1, name: "Red Sports Tee",...)]
  0: {id: "1606771604", quantity: 1, basketQuantity: 1, name: "Red Sports Tee",...}
    basketQuantity: 1
    brand: "Hero"
    category: "Top"
    currency: ""
    id: "1606771604"
    image: "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/womens-red-t-shirt_925x_b78d289c-e4d9-4b00-bde3-671c43b03841_small.jpg?v=1606771604"
    location: "https://the-apparel-store-by-hero.myshopify.com/products/red-sports-tee"
    name: "Red Sports Tee"
    price: 50
    quantity: 1
    variant: "Red Sports Tee / Red"
    type: "ecommerce:add"
  
```

basketQuantity incorrectly passed into the Basket Add Event

This value should match the current quantity of products in the shopping basket


JUST ADDED TO YOUR CART ✕



Red Sports Tee Qty: 1
Title: Red Sports Tee
Color: Red

VIEW CART (4)

[Continue shopping](#)



Red Sports Tee
\$50.00

Title
Red Sports Tee

Color
Red

ADD TO CART

BUY IT NOW

2 / 107

Console What's New

tracking

5000 ms 10000 ms 15000 ms 20000 ms 25000 ms 30000 ms 35000 ms

Referrer Policy: strict-origin-when-cross-origin

Response Headers (18)

Request Headers (18)

Request Payload view source

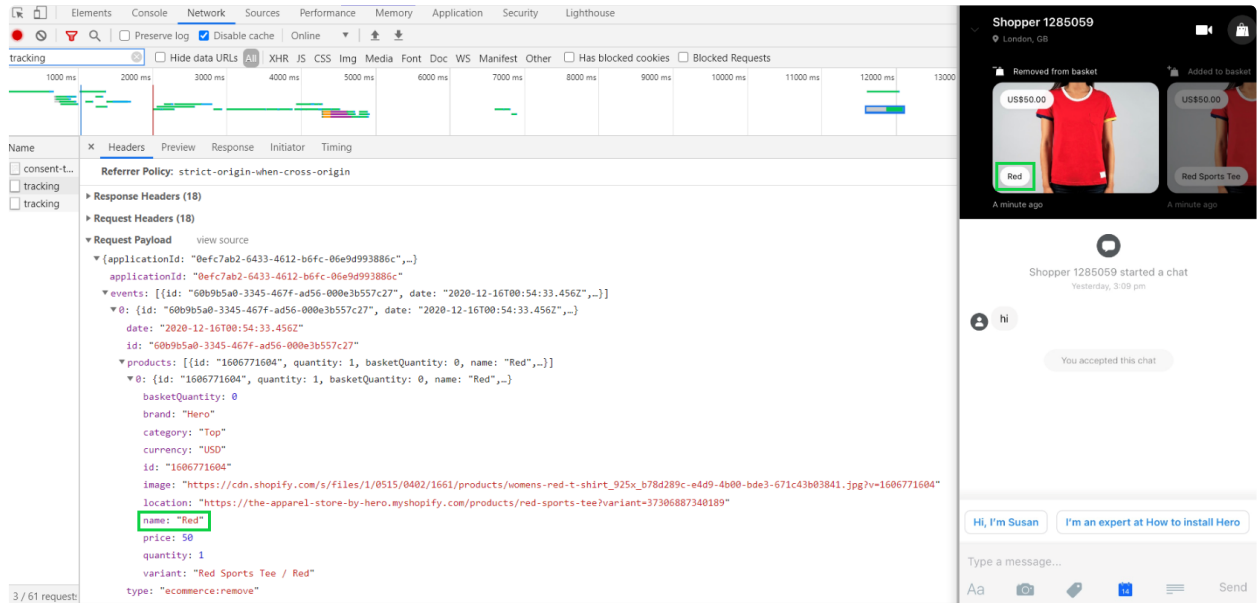
```
{applicationId: "0efc7ab2-6433-4612-b6fc-06e94993886c", events: [...], applicationId: "0efc7ab2-6433-4612-b6fc-06e94993886c"}
events: [...]
0: {id: "09a93aaf-686a-4d4c-8cb6-9123027d91e0", date: "2020-12-18T03:52:24.046Z", type: "ecommerce:add",...}
  date: "2020-12-18T03:52:24.046Z"
  id: "09a93aaf-686a-4d4c-8cb6-9123027d91e0"
  products: [{"id": "1606771604", quantity: 1, basketQuantity: 0, name: "Red Sports Tee",...}]
    0: {id: "1606771604", quantity: 1, basketQuantity: 0, name: "Red Sports Tee",...}
      basketQuantity: 0
      brand: "Hero"
      category: "Top"
      currency: "USD"
      id: "1606771604"
      image: "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/womens-red-t-shirt_925x_b78d289c-e4d9-4b08-bde3-671c43b03841_small.jpg?v=1606771604"
      location: "https://the-apparel-store-by-hero.myshopify.com/products/red-sports-tee"
      name: "Red Sports Tee"
      price: 50
      quantity: 1
      variant: "Red Sports Tee / Red"
      type: "ecommerce:add"
  type: "ecommerce:add"
```

Basket Remove

Shopper Event Troubleshooting

product name displays incorrectly on Basket Remove Event card

If an incorrect product name is displaying on the Shopper Event card, then this would mean that the name provided in the Shopper Event does not match the actual name of the product. It is important to ensure that the name value implemented in the event matches the actual name of the product.

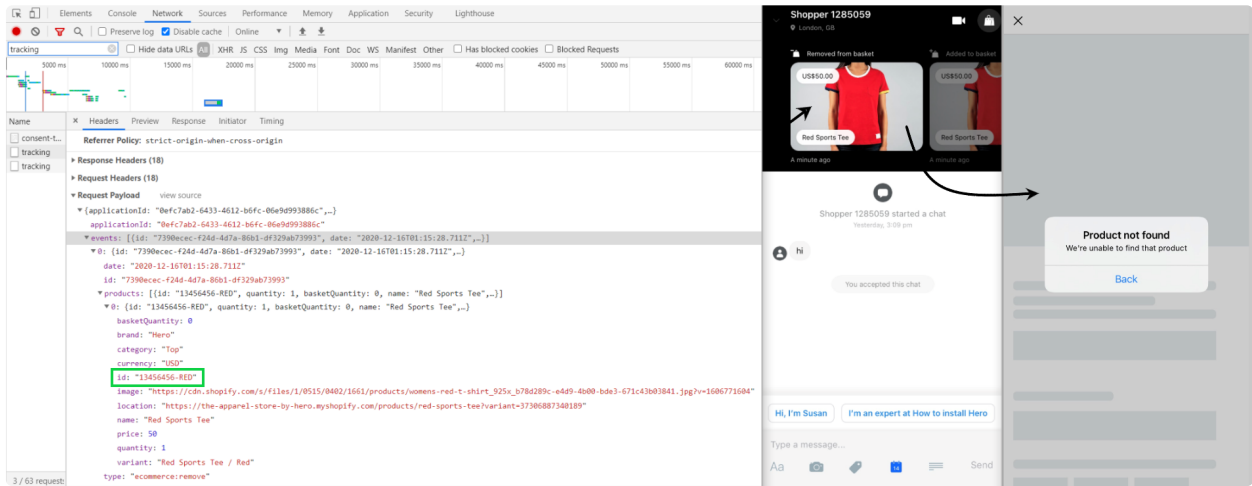


Incorrect name provided in Shopper Event

'Product not found' message appears on the Store App when a user taps on a Basket Remove Event card

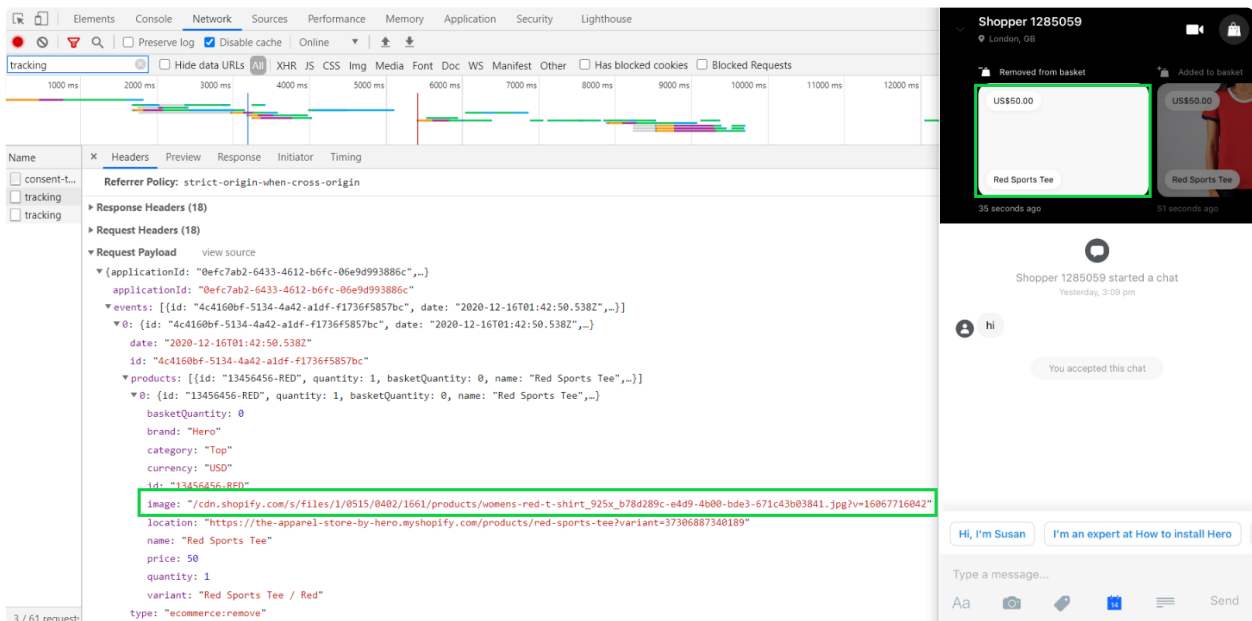
Assuming that the product feed has been indexed and products are displaying in the product catalogue, if this issue is present, then this would mean that **the product ID (SKU or variant ID) provided in the Shopper Event, does not match the product ID provided in the Product Feed** for that specific product.

For example, the unique product ID / SKU provided in the feed for the below product Red Sports Tee is 1606771604 and the ID provided in the event 1606771604-RED does not match this exact value, then this will result in the 'Product not found' error message when attempting to view the product detail page.



Product image is missing from Basket Remove Event card

This issue displayed on the app would occur if the image field / the image URL value is missing within the product array or if the URL provided is not https and / or is not a valid URL. All three scenarios would result in the product image to not display as expected on the app for the Shopper Event card.



Product Image not displaying on Store App due to image URL not being https.

Price on Basket Remove Event card does not match the current price of the product

If the price displayed on the shopper event card does not match the actual / current price of the product being tracked, then this would mean that the price provided within the product array of the Shopper Event, does not match the current price of the product.

It is important to ensure that the product price within the event matches the current price of the product provided on your website and also in your Product Feed.

The screenshot shows a network request for a 'Basket Remove' event. The request payload includes the following data:

```

{
  applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c",
  applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c",
  events: [
    {
      id: "6ecd870f-2970-4bbc-8d86-de676adafc74",
      date: "2020-12-16T02:49:37.107Z",
      id: "6ecd870f-2970-4bbc-8d86-de676adafc74",
      products: [
        {
          id: "1606771604",
          quantity: 1,
          basketQuantity: 0,
          name: "Red Sports Tee",
          basketQuantity: 0,
          brand: "Hero",
          category: "Top",
          currency: "USD",
          id: "1606771604",
          image: "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/womens-red-t-shirt_925x_b78d289c-e4d9-4b00-bde3-671c43b03841.jpg?v=1606771604",
          location: "https://the-apparel-store-by-hero.myshopify.com/products/red-sports-tee?variant=37306887340189",
          name: "Red Sports Tee",
          price: 0.05,
          quantity: 1,
          variant: "Red Sports Tee / Red",
          type: "ecommerce.remove"
        }
      ]
    }
  ]
}

```

The mobile app screenshot shows a notification: "Removed from basket" with a price of US\$0.05 and a "Red Sports Tee" image.

Price value is missing from Basket Remove Event card

If the price value is missing from the Shopper Event card, this would mean that the price value passed in the 'Basket remove' Shopper Event has been provided as a string and not a number.

The screenshot shows a network request for a 'Basket Add' event. The request payload includes the following data:

```

{
  applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c",
  applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c",
  events: [
    {
      id: "d4282ac0-1cd0-4423-b24b-80e87944ef66",
      date: "2020-12-16T03:15:02.940Z",
      id: "d4282ac0-1cd0-4423-b24b-80e87944ef66",
      products: [
        {
          id: "1606771604",
          quantity: 1,
          basketQuantity: 0,
          name: "Red Sports Tee",
          basketQuantity: 0,
          brand: "Hero",
          category: "Top",
          currency: "USD",
          id: "1606771604",
          image: "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/womens-red-t-shirt_925x_b78d289c-e4d9-4b00-bde3-671c43b03841.jpg?v=1606771604",
          location: "https://the-apparel-store-by-hero.myshopify.com/products/red-sports-tee?variant=37306887340189",
          name: "Red Sports Tee",
          price: 50.00,
          quantity: 1,
          variant: "Red Sports Tee / Red",
          type: "ecommerce.remove"
        }
      ]
    }
  ]
}

```

The mobile app screenshot shows a notification: "Added to basket" with a price of US\$50.00 and a "Red Sports Tee" image.

It is important to ensure that price, quantity and basketQuantity values are implemented as a number and not a string in the Shopper Event.

Location URL is missing / is not provided as https

THE APPAREL STORE BY HERO

Your cart
Your cart is currently empty.
CONTINUE SHOPPING →

Quick links
Search
Newsletter

Network: tracking
 Name: tracking
 Headers: sec-fetch-dest: empty, sec-fetch-mode: cors, sec-fetch-site: cross-site, user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
 Request Payload: {applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c",...}, events: [{"id": "131e17f9-4838-4055-98cc-e0276101c1f", date: "2020-12-21T18:42:55.981Z",...}], products: [{"id": "1606771604", quantity: 1, basketQuantity: 0, name: "Red Sports Tee",...}], location: "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/womens-red-t-shirt_925x_b78d289c-e4d9-4b00-bde3-671c43b03841.jpg?v=1606771604", name: "Red Sports Tee", price: 50, quantity: 1, variant: "Red Sports Tee / Red", type: "ecommerce:remove"}

currency field or currency value is missing within the product array

THE APPAREL STORE BY HERO

Your cart
Your cart is currently empty.
CONTINUE SHOPPING →

Quick links
Search
Newsletter

Network: tracking
 Name: tracking
 Headers: sec-fetch-dest: empty, sec-fetch-mode: cors, sec-fetch-site: cross-site, user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36
 Request Payload: {applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c",...}, events: [{"id": "f68f2183-222a-41d9-9674-9b52a053c3bb", date: "2020-12-21T18:37:39.493Z",...}], products: [{"id": "1606771604", quantity: 1, basketQuantity: 0, name: "Red Sports Tee",...}], currency: " ", id: "1606771604", image: "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/womens-red-t-shirt_925x_b78d289c-e4d9-4b00-bde3-671c43b03841.jpg?v=1606771604", name: "Red Sports Tee", price: 50, quantity: 1, variant: "Red Sports Tee / Red", type: "ecommerce:remove"}

Incorrect basketQuantity passed into the Shopper Event

This value should match the current quantity of products in the shopping basket.

The screenshot displays the front-end of 'THE APPAREL STORE BY HERO' and its network layer. On the left, the cart page shows 'Your cart' with the message 'Your cart is currently empty.' and a 'CONTINUE SHOPPING' button. The right side shows the browser's developer tools with the Network tab selected. A request for 'tracking' is expanded, showing headers and a request payload. The payload includes a list of products, with one product having a 'basketQuantity' of 1. The text 'basketQuantity: 1' is highlighted in green in the original image.

```
sec-fetch-dest: empty
sec-fetch-mode: cors
sec-fetch-site: cross-site
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36

▼ Request Payload
  applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c",...
  applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c"
  events: [{"id": "21d2a3e8-8409-4bc8-8eab-427675119b47", date: "2020-12-21T18:29:22.414Z", ...}]
  0: {id: "21d2a3e8-8409-4bc8-8eab-427675119b47", date: "2020-12-21T18:29:22.414Z", ...}
  id: "21d2a3e8-8409-4bc8-8eab-427675119b47"
  products: [{"id": "1606771604", quantity: 1, basketQuantity: 1, name: "Red Sports Tee", ...}]
  0: {id: "1606771604", quantity: 1, basketQuantity: 1, name: "Red Sports Tee", ...}
  basketQuantity: 1
  brand: "Hero"
  category: "Top"
  currency: "USD"
  id: "1606771604"
  image: "https://cdn.shopify.com/s/files/1/0515/0402/1661/products/womens-red-t-shirt_925x_b78d289c-e4d9-4b00-bde3-671c43b03841.jpg?v=1606771604"
  location: "https://the-apparel-store-by-hero.myshopify.com/products/red-sports-tee?variant=37306887340189"
  name: "Red Sports Tee"
  price: 50
  quantity: 1
  variant: "Red Sports Tee / Red"
  type: "ecommerce/remove"
```

Here, the basketQuantity is incorrect as it does not match the current quantity in shopping basket.

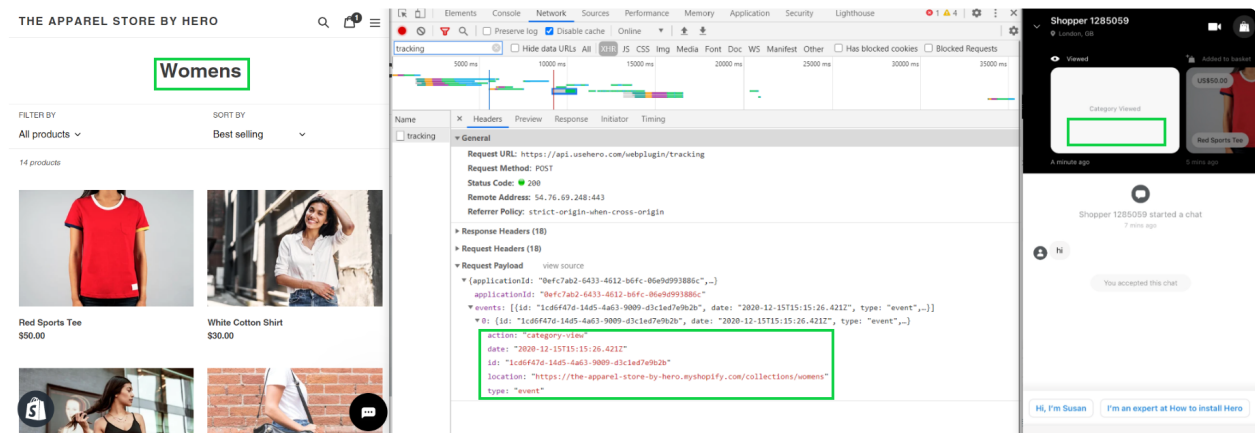
Category View

Shopper Event Troubleshooting

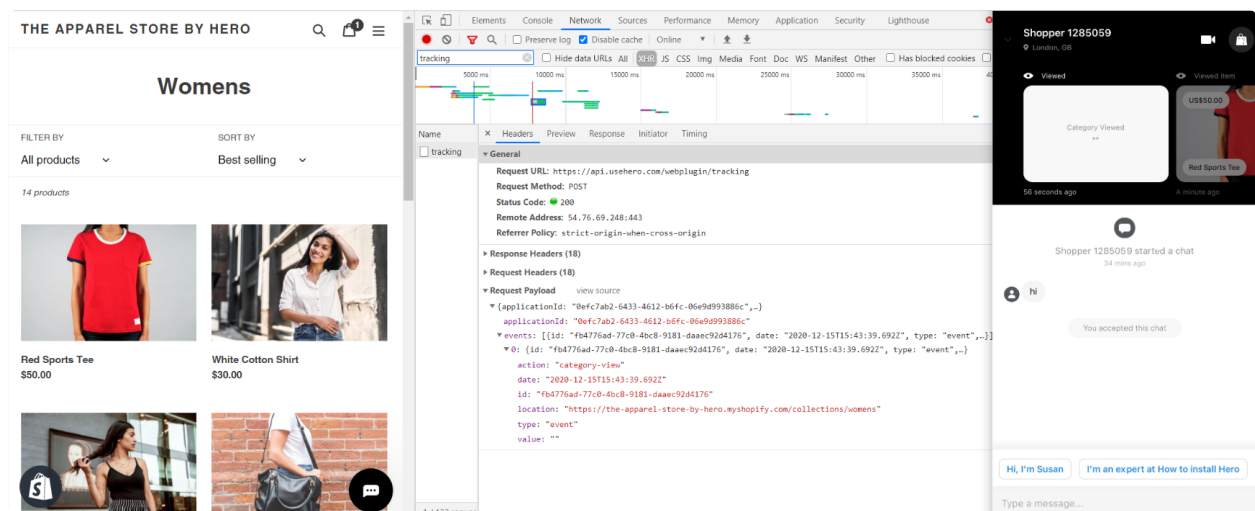
Category View value not showing on the Store App despite event firing in the request payload

If the 'Category View' value is missing on the 'Category View' Shopper Event card in the Store App, then this would mean that the value field is missing or empty or the value has not been provided as a string for the Shopper Event.

Please ensure that this field has been passed into the event and in the format of a string to ensure that it displays as expected on the Store App.



Category view value not shown on Store App due to value field not provided in Shopper Event.



'Category view' card shows as empty because the value field is empty in the Shopper Event.

Location URL is missing / is not provided as https

Mens

FILTER BY: All products | SORT BY: Best selling

6 products



Chequered Red Shirt
\$50.00



Ocean Blue Shirt
\$50.00



LED High Tops
\$80.00



Blue Silk Tuxedo
\$70.00



Elements Console Network Sources Performance Memory Application Security Lighthouse

tracking

1000 ms 2000 ms 3000 ms 4000 ms 5000 ms 6000 ms 7000 ms 8000 ms 9000 ms 10000 ms 11000 ms 12000 ms

Name: tracking

General

Request URL: https://api.usehero.com/webplugin/tracking

Request Method: POST

Status Code: 200

Remote Address: 54.76.69.248:443

Referrer Policy: strict-origin-when-cross-origin

Response Headers (18)

Request Headers (18)

Request Payload

```
{applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c",...}
  applicationId: "0efc7ab2-6433-4612-b6fc-06e9d993886c"
  events: [{id: "70f4d15d-7623-41dd-8bbd-e174d17460f1", date: "2020-12-18T09:00:36.200Z", type: "event",-}]
  0: {id: "70f4d15d-7623-41dd-8bbd-e174d17460f1", date: "2020-12-18T09:00:36.200Z", type: "event",-}
    action: "category-view"
    date: "2020-12-18T09:00:36.200Z"
    id: "70f4d15d-7623-41dd-8bbd-e174d17460f1"
    location: ""
    type: "event"
    value: "Mens"
  }
```

1 / 70 requests

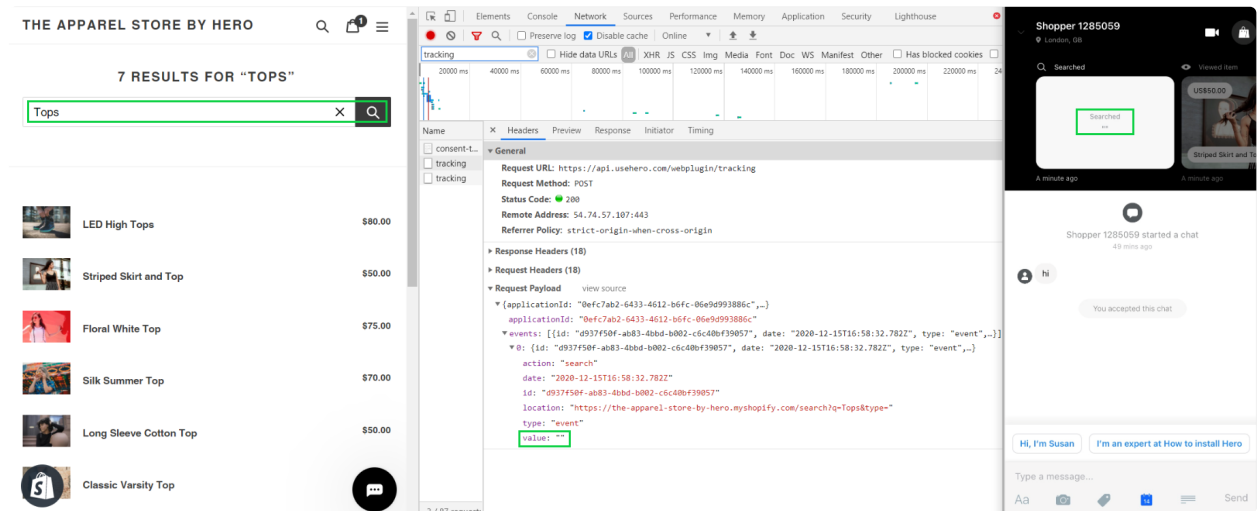
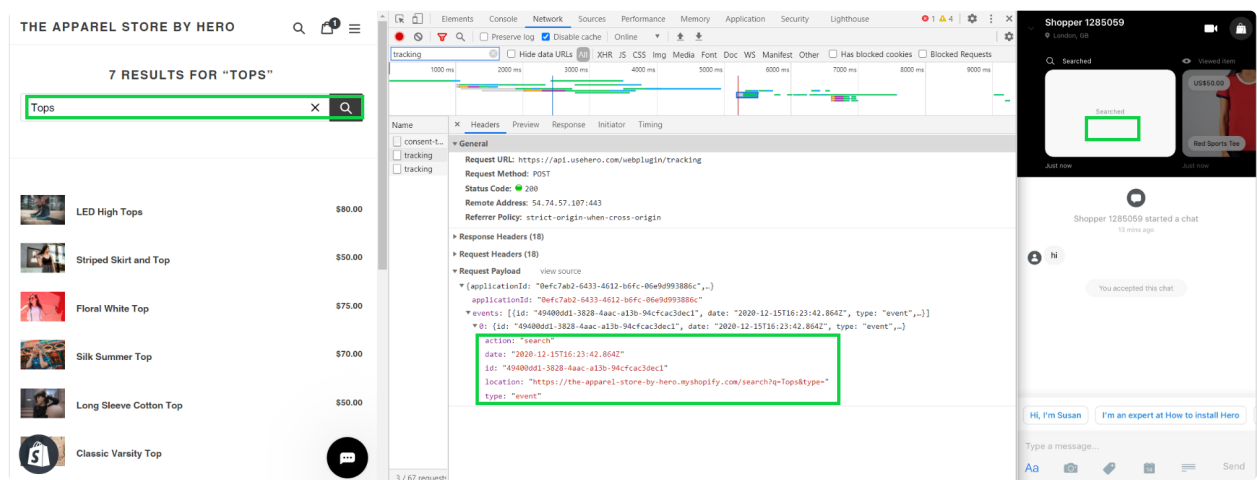
Search

Shopper Event Troubleshooting

'Search' value not displaying on the Store App despite event firing in the request payload

If the 'Search' value is missing on the 'Search' event card on the Store App, then this would mean that the value field within the Shopper Event is missing or empty or the value has not been provided as a string.

Please ensure that this field has been passed into the event and in the format of a string to ensure that it displays on the Store App.



Here the 'Search' card shows an empty value because the value field is empty in the event.

Wrong Shopper Event captured on the Store App

If the wrong Shopper Event is captured on the Store App this would mean that the Shopper Event code has been implemented to the wrong pages of your website.

For example, if the 'Search' event is triggered when a customer clicks on a category this would mean that the 'Search' Shopper Event code has been implemented to a page with a category structure and not for a 'Search' shopper action. This would cause the event to fire incorrectly for a different Shopper action.

The screenshot shows a web browser displaying 'THE APPAREL STORE BY HERO' with a 'Womens Category' filter. A network tracking tool is open, showing a POST request to 'https://api.usehero.com/webplugin/tracking'. The request payload contains an event with the following details:

```
{
  "action": "Search",
  "date": "2020-12-15T17:32:51.844Z",
  "id": "92056deb-3369-44b5-8faa-2209d69da998",
  "location": "https://the-apparel-store-by-hero.myshopify.com/search?q=bltype",
  "type": "event"
}
```

The tracking tool also shows a chat interface for 'Shopper 1285059' with a 'Searched' message.

Here you can see a Search event firing for a different shopper action.

Location URL is missing/is not provided as https

The location field should be passed into the Shopper Event with a valid URL and this URL should be https .

The screenshot shows a web browser displaying 'THE APPAREL STORE BY HERO' with search results for 'SHIRTS'. A network tracking tool is open, showing a POST request to 'https://api.usehero.com/webplugin/tracking'. The request payload contains an event with the following details:

```
{
  "action": "Search",
  "date": "2020-12-18T04:30:59.734Z",
  "id": "d8fff243-1fc2-4176-9de1-b4ada8a050ba",
  "location": "",
  "type": "event",
  "value": "Shirts"
}
```

The tracking tool also shows a list of search results for 'SHIRTS' including 'White Cotton Shirt', 'Chequered Red Shirt', 'Ocean Blue Shirt', and 'Red Sports Tee'.




Ready Stores

Recommended WiFi speeds

Ensure access to good WiFi in stores

To provide the best possible associate experience, we recommend retailers have access to good WiFi coverage in their stores.


The demands on the connection when using Virtual Shopping will vary depending on the activity taking place:

-  **Light** On-site chats, sharing product recommendations or sending invites
-  **Medium** Sending images and then video
-  **Heavy** Video streaming

Virtual Shopping will work on a 'good' 3G connection for both shoppers on your website and associates using the Store app.

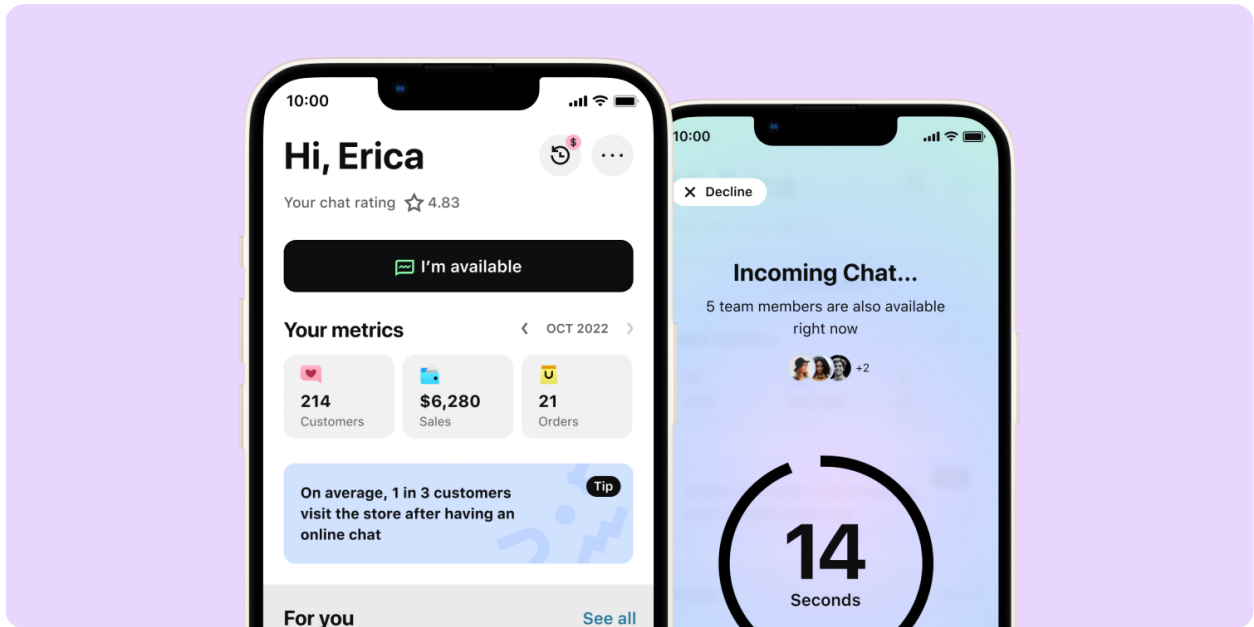
Here is a breakdown of what Virtual Shopping tests on and the expected experience:

Connection	Download	Upload	Experience
Good 3G	1.5 Mb/s	750 kB/s	Okay
Regular 4G/LTE	4.0 Mb/s	3.0 mB/s	Better
WiFi	30 Mb/s	15.0 mB/s	Best


 To find out how many simultaneous video streaming calls your network can support, run a test [here](#).

Supported devices

The Store App supports iOS and Android devices 



The Store App for associates is available on iOS from the [Apple App Store](#) or Android from the [Google Play Store](#).

 You should ensure your chosen devices conform to the minimum requirements listed below.


System requirements

For the best performance, we recommend using the Store app on the most recent devices and operating systems.

iOS

The Store app supports either iPhones or iPads.

- **Minimum device requirements:** iPhone 7 / iPad 7th Generation
- **Minimum operating system:** iOS 13 or above

 The Store App will run on iPad, the app has a mobile-first design and so will only display in portrait mode.

Please note the Store App does not support the use of iPods

Android

For the best experience with Store App on Android, we recommend using latest model Android devices.

- **Minimum operating system:** OS 11.0 or above

Centrally provisioned devices

If your devices are provisioned by a central program please ensure that:

- ✓ The app is set to auto-update
- ✓ Push Notifications are enabled
- ✓ [Ports and Domains](#) are whitelisted

Ports and Domains

Ensure Virtual Shopping will function on your corporate WiFi

If you blacklist any ports or domains on your corporate WiFi then ports and domains listed here must be whitelisted for Virtual Shopping to function correctly on your team's devices when they are connected to this network.

Virtual Shopping Core Product

Action	Where	Why
Whitelist Domain	<code>*.usehero.com</code>	Key API to ensure functionality across the application
Whitelist Domain	<code>*.virtual-shopping.klarna.com</code>	Key API to ensure functionality across the application (Websockets)
Whitelist Domain	<code>*.twilio.com</code> <code>tsock.twilio.com</code> <code>tsock.us1.twilio.com</code> <code>tsock.ie1.twilio.com</code> <code>media.us1.twilio.com</code> <code>mcs.us1.twilio.com</code>	Real time messaging and video support across the application
Whitelist Domain	<code>*.mixpanel.com</code> <code>mixpanel.com</code>	Reporting and analytics
Whitelist Domain	<code>hero-prod-assets.s3-eu-west-1.amazonaws.com</code>	General Storage across the application - required for core functionality i.e. images, video avatars etc...

* Domains that also need to be whitelisted on https

Apple Push Notification Service


Action	Where	Why
Whitelist IP and Port	IP address block <code>17.0.0.0/8</code> on ports <code>TCP 5223</code> and <code>TCP 443</code>	Push notifications across all devices

Google Cloud Messaging

Action	Where	Why
Whitelist Port	Ports: 5228 5229 5230	Push notifications across Android Devices * please note there is no fixed address: https://firebase.google.com/docs/cloud-messaging/concept-options#messaging-ports-and-your-firewall

Live Video Streaming

For Live Video Streaming, please follow [Twilio's recommended whitelisting](#).

 We recommend the following but we would ask you to verify using [Twilio's testing link](#) to ensure correct setup.

To run the test, access the above link from a device currently connected to the store network from where you are going to be using Virtual Shopping. For the most accurate results, your test device should have the same configuration as the 'Go Live; store associate's device.

Region ID	Location	Host Name	Port and Protocol
gll	Global Low Latency (default)	global.vss.twilio.com	443 WSS
	Additional Connectivity requirement	sdkgw.us1.twilio.com	(WSS/443)
	Additional Connectivity requirement	ecs.us1.twilio.com	(HTTPS/443)

Product Feed indexing

If your product feed is protected with an IP address allow list, please add the following IP addresses to allow our platform to fetch the feed:

IP address
54.228.73.53
54.78.118.192
99.81.185.60
54.171.103.32

App deletion requests

If you receive an email notification that a team member has requested for their Store App account be deleted, here is how to delete their access and reassign their contacts.

Store team members can request that their Store App account be deleted from the profile page on their app.

Once they do, you will be notified and asked to complete the following actions:

1. Deactivate the anonymized account.
2. Reassign any contacts associated with the anonymized account as needed.

How to locate anonymized accounts

Visit "Manage team" in your Dashboard and look for:

Name: Anonymised (DATE)

Email Address: user.xxxxx@vs-anon.klarna.com


Manage Team

Refresh ↻

Approve and manage team members

Pending (0) **Approved (1)** Deactivated (63)

Sort By Name ▾ Anonymised 2022-05-24 🔍 Showing 1-1 of 1 < >

Team Members	Store	Team(s)	Profile	Academy	Demo	Status
 Anonymised 2022-05-24 Registered 18 Sept 2020	1214 Regent Street L...	Bags + 6 others	✓	✓	✓	Approved ▾

Anonymised 2022-05-24
• Not yet active
user.Q1Dfpm7POL@vs-anon.klarna.com

Give Anonymised access to Contacts?

Add Contacts +

Problems with Conversations?

End Current Conversation

Reset Layer Username


Showing 1-1 of 1 < >

FAQs

What Tracking Technologies does Virtual Shopping use?

Tracking Types

Virtual Shopping On-site Chat stores a series of cookies and local storage that help us to understand who a user is, surface their chat history, maintain the correct state of interaction with the on-site chat, correctly attribute any sales to the product expert who helped them and manage experiments.


 We do not utilise third-party cookies for tracking. Klarna sets first party client-side cookies.

Here are the Virtual Shopping tracking technologies currently in use:

Session

This stores data related to the user session, to ensure we can create new conversations and load conversation history within On-site Chat.


name	hero-session-* (* is replaced by your application ID)
contains	User token and shopper's first name (if entered)
expiry	1 year
type	cookie

 The Session cookie is not configurable and cannot be deactivated as it is essential for providing core messaging functionality.

State

This stores data related to the user's current state, to allow us to understand what current interactions the user has had in the On-site Chat i.e. whether it is open or closed.


name	hero-state-* (* is replaced by your applica ID)
contains	Current Messenger state
expiry	1 year
type	local storage

 The State cookie is not configurable and cannot be deactivated as it is essential for manage the users state and ensure a smooth and consistent shopper experience.

Attribution

The attribution cookie is always set but only contains information when Employee ID is enabled as a step in product expert registration. When a shopper is or has been connected to an associate, this cookie will be populated with the Associate ID, so that they can be attributed accurately. The cookie is set when moving to a new page.

name	hero-associate-data
contains	Associate employee ID
expiry	1 year
type	cookie

 The Attribution cookie cannot be deactivated, but will only be populated where an employee ID is held for the product expert

Shopper Event tracking

The Shopper Event tracking local storage is used to store shopper events before they are sent to the server.

name	hero-tracking:* (* is replaced by your application ID)
contains	Shopper Events
expiry	Not Set
type	local storage

Experiment tracking

The Experiment entry in local storage is used to store experiment and variant IDs when an A/B is running.

name	hero-experiment:* (* is replaced by your application ID)
contains	Experiment and variant ID when an A/B test is running
expiry	Not Set
type	local storage

Multi-tab support

The Multi-tab support uses an entry in session storage to ensure cross-tab support for conversations and video calls.

name	hero-tab-id
contains	Generated tab ID
expiry	Expires when the tab/browser is closed
type	session storage

User ID (deprecated)

The User ID cookie stores the Virtual Shopping Shopper ID and is accessible by the website to support integrating Virtual Shopping data into the retailer's own reporting.

name	hero-user-id
contains	User ID
expiry	1 month
type	cookie



The **Virtual Shopping User ID cookie** will soon be deprecated. If you are using this cookie then please get in touch with help.virtualshopping@klarna.com to discuss migration plans.

Further questions

What are the recommended server configurations to support Virtual Shopping cookies?

Configure Your server, NGINX and/or Apache to accept headers of size 8Mb - 16Mb

What is the expected cookie size?

As soon as On-site Chat loads a 130 byte cookie is set. Once a shopper starts a chat a 636 byte cookie is set. This is the largest expected size of the cookie.

How big is the Javascript file and when is it loaded?

The On-site Chat JavaScript is designed to offer maximum functionality with minimum impact on your overall site performance.

The Virtual Shopping On-site Chat JavaScript loads asynchronously for optimal performance, ensuring all priority website content is loaded before the Virtual Shopping JavaScript starts to load. It also loads only the parts that are needed depending on the user interaction. The following details what is loaded in each scenario.

The Loader is a JavaScript file which handles the loading of the On-site Chat launcher button and messenger. It's as small as possible whilst still allowing us to understand display rules, determine sales attribution and understand the shopper journey to inform the product expert. On a page where Virtual Shopping won't show to a shopper, such as a checkout page, this is all that loads.

Total load: 27 KB

On pages where the customer can interact with the On-site Chat, there are three exclusive scenarios which determine the plugin size:

The Launcher and Launcher Bubble loads but the user does not interact with the plugin.

Total load: 275 KB

The Launcher is interacted with to open the Messenger.

Total load: 340 KB

The Launcher is interacted with to open the Messenger and the customer starts a conversation.

Total load: 445 KB

There are a number of other minor asynchronous loads beyond this, but these are used by a minimum number of users for specific functions around map and country selector functionality. These are kept to a minimum.

We are continually committed to reducing the file size where possible.

Which languages are supported?

The On-site Chat plugin and Store App are localized into the following languages:

- English (US)
- Chinese (simplified)
- Chinese (traditional)
- Danish
- Dutch
- Finnish
- French
- German
- Italian
- Japanese
- Korean
- Norwegian
- Polish
- Portuguese
- Spanish
- Swedish



The language shown will depend on the languages selected in the shopper's browser settings for the Plugin or the product expert's mobile device for the App.

What happened to Hero?

Hero became [Klarna Virtual Shopping](#) in March 2022.

Do I need to update anything on my site?

No, everything will continue to work the same as it did before. There's no dependency on you to update what it is deployed to your site.

What's changed with the Hero App?

The Hero app is now called the Store App. It's under the same listing and so you will continue to receive app updates as normal. From an architectural or technical perspective nothing has changed, it's fundamentally the same app; just with a new look and feel. The only change is that the app is now published by Klarna instead of Hero.

What does this mean for how you handle data?

Nothing has changed. Data is collected, stored and processed in the same way.

What happens when Virtual Shopping is deactivated?

60 days after an instance of Virtual Shopping is officially deactivated, all PII data associated with the deactivated instance will automatically be anonymised.

Within this 60 day time frame, we will provide our merchants with access to this data via a secure method of collection.



Please note that it is not possible to access this data after the 60 day period.