

Tutorials

- [How do I add featured products to the homepage of my store?](#)
- [How do I add a contact form to my storefront?](#)
- [How do I add a header image to the listings page?](#)
- [How can I add a tax toggle to my store?](#)
- [How can I add a cart panel to my store?](#)

How do I add featured products to the homepage of my store?

The recommended way to add featured products to the homepage of your store is to use the [aerocargo/listing-collections](#) package.

Install

Run this command in the root of your project:

```
composer require aerocargo/listing-collections
```

Create a section

1. Create the file:

```
themes/acme/resources/sections/best-sellers.twig
```

Replace `acme` with the name of your theme.

2. Add the twig:

```
{% set collection = listings([{"tag_name.en": "Best Seller"}], 8) %}

{% component "listings" with { listings: collection } csr %}
<div class="aero-listing-collection">
    <h3 class="aero-listing-collection__title">Best Sellers</h3>
```

```
<ul class="aero-listings grid grid-cols-4">
  <li v-for="listing in listings.data" class="aero-listing">
    {% snippet "listing-card" %}
  </li>
</ul>
</div>
{% endcomponent %}
```

Use the section

Reference your section where you'd like it to display and pass through the variables, here we'll use the tag name `best-sellers` this will be referenced in the next step.

```
{% section "best-sellers" %}
```

Now products tagged with `best-sellers` will be rendered wherever our section is used.

See more info in the [aerocargo/listing-collections docs](#).

How do I add a contact form to my storefront?

1. Create the file:

```
themes/acme/resources/forms/contact.twig
```

Replace `acme` with the name of your theme.

Any files created inside the `forms` directory can be included on a page using `{% form "contact" %}`.

2. Add the twig:

```
<form action="{{ route('form', 'contact') }}" method="post">
  {{ csrf_field() }}
  <div>
    <label for="name">Name:</label>
    <input id="name" type="text" name="name" autocomplete="name" required
placeholder="First Name" value="{{ old('name') }}">
  </div>
  <div>
    <label for="name">Email Address:</label>
    <input id="email" type="text" name="email" autocomplete="email" required
placeholder="Email Address" value="{{ old('email') }}">
  </div>
  <div>
    <label for="name">Your Message</label>
    <textarea id="message" type="text" name="message" autocomplete="off" required
placeholder="Your message">{{ old('message') }}</textarea>
  </div>
  <div>
    <button type="submit">Submit</button>
  </div>
```

```
</form>
```

3. Display a session success message:

```
{% if session('success') %}  
  <div role="alert">  
    <span>Your message has been sent.</span>  
  </div>  
{% endif %}
```

Mail Notification

You can create a mail notification in the admin for the `Form Submitted` event (leave `value` empty to target all forms, or put `contact` to target this one specifically)

Then add this to the body of the mail notification, to output the submission:

```
{% for field, value in fields %}  
  <p>{{ field }}: {{ value }}</p>  
{% endfor %}
```

How do I add a header image to the listings page?

On a listings page we can pass through an image from the admin using the `combination` variable in the `listings.twig` view.

The combination is the data model that holds the information for the particular listings page being viewed - since listings can be made up of a category, a manufacturer or both.

We'll output the entire image block, passing widths & heights for each size:

```
{{ combination.image_block.render({
  sizes: {
    small: {
      width: 360,
      height: 360,
    },
    medium: {
      width: 1024,
      height: 540,
    },
    large: {
      width: 1920,
      height: 900,
    },
  },
}) | raw }}
```

Optionally, we can add some styling to our header to create a nice component for our theme:

```
<div class="aero-listing-header flex flex-col lg:flex-row gap-4 lg:gap-8">
  <div class="aero-listing-header__image w-full lg:flex-1">
    {{ combination.image_block.render({sizes: {small: {width: 360, height: 360}, medium:
{width: 1024, height: 540}, large: {width: 1920, height: 900}}}) | raw }}
  </div>
  <div class="aero-listing-header__content">
```

```
<h1 class="aero-listing-header__title text-xl lg:text-3xl">
  <template v-if="search_term">Search for "{{{ search_term }}" }}</template>
  <template v-else>{{{ heading }}" }}</template>
</h1>
</div>
</div>
```

How can I add a tax toggle to my store?

First, install `aerocargo/tax-toggler`:

```
composer require aerocargo/tax-toggler
```

This module adds a `tax_state()` twig function that returns `inc` or `ex` depending on configuration, see [here](#) for more information.

Next let's create the `layouts/partials/tax-toggle.twig` file:

```
<label class="inline-flex items-center cursor-pointer">
  <span class="hidden md:flex text-xs font-medium uppercase text-gray-900 mr-2">
    <span class="price--inc">inc VAT</span>
    <span class="price--ex">ex VAT</span>
  </span>
  <input type="checkbox" id="tax-toggler" value class="sr-only peer" {{ tax_state() == "inc"
? "checked" : "" }}>
  <span class="relative w-9 h-5 bg-gray-200 peer-focus:outline-none rounded-full peer peer-
checked:after:translate-x-full peer-checked:after:border-white after:content-['']
after:absolute after:top-[2px] after:start-[2px] after:bg-white after:border-gray-300
after:border after:rounded-full after:h-4 after:w-4 after:transition-all"></span>
</label>

{% js %}
  <script>
    (() => {
      let state = "{{ tax_state() }}"

      function togglePrices(state) {
        let date = new Date();
        date.setTime(date.getTime() + 2592000000);
        document.body.classList.remove(state === 'inc' ? 'prices--ex' : 'prices--
inc');
```

```

        document.body.classList.add(state === 'inc' ? 'prices--inc' : 'prices--ex');
        document.cookie = `prices=${state}; path=/; expires=${date.toUTCString()}`;
    }

    togglePrices(state);

    let toggle = document.getElementById('tax-toggler');

    toggle.checked = state === 'inc';

    toggle.addEventListener('change', () => {
        togglePrices(toggle.checked ? 'inc' : 'ex');
    });
    })();
</script>
{% endjs %}

{% css %}
<style>
    body.prices--inc .price--inc, body.prices--ex .price--ex { display: initial; }
    body.prices--inc .price--ex, body.prices--ex .price--inc { display: none; }
</style>
{% endcss %}

```

`2592000000` is 30 days in milliseconds.

Then let's include this file in our `header.twig`:

```
{% include "layouts/partial/tax-toggle.twig" %}
```

Now we just have to mark any prices we display to use this toggle, for example in `product.twig`:

```

{# Before #}
<span class="text-xl">{{ "{{ sale_price.inc }}" }}</span>

{# After #}
<span class="text-xl price--inc">{{ "{{ sale_price.inc }}" }}</span>
<span class="text-xl price--ex">{{ "{{ sale_price.ex }}" }}</span>

```


How can I add a cart panel to my store?

This tutorial will provide the code needed to setup a cart panel slide-out system for your store.

First, lets add the `layouts/partials/cart-panel.twig` code:

```
<div id="cart-panel" class="cart-panel isolate fixed z-[999] bottom-0 right-0 max-h-full
overflow-y-auto overscroll-contain h-[90%] md:h-dvh w-full md:max-w-2xl bg-white [body.cart-
panel--open_&]:shadow-[0_0_50px_0_rgba(0,0,0,0.3)]">
  <button type="button" class="absolute z-10 right-0 top-0 flex items-center justify-center
p-3 m-4" onclick="document.getElementById('cart-panel-cover').click()" aria-label="Close
cart">
    <svg viewBox="0 0 24 24" fill="currentColor" class="size-6">
      <path fill-rule="evenodd" d="M5.47 5.47a.75.75 0 0 1 1.06 0L12 10.94l5.47-
5.47a.75.75 0 1 1 1.06 1.06L13.06 12l5.47 5.47a.75.75 0 1 1-1.06 1.06L12 13.06l-5.47
5.47a.75.75 0 0 1-1.06-1.06L10.94 12 5.47 6.53a.75.75 0 0 1 0-1.06Z" clip-rule="evenodd" />
    </svg>
  </button>
  <div class="relative px-4 pt-6 lg:pt-12 pb-6 lg:pb-12 mx-auto min-h-full w-full max-w-xl
items-center text-center">
    <h2 class="copy copy-heading text-base lg:text-lg font-medium">My Cart</h2>

    <div class="pt-2.5 md:pt-3.5 lg:pt-6 pb-2.5 md:pb-3.5 lg:pb-6">
      {% if cart().items is empty %}
        <div class="space-y-4">
          <p class="copy copy-minor-heading leading-6 text-sm">Your Shopping Cart is
Empty</p>
          <div class="copy copy-text text-sm leading-6 [&a]:underline">You haven't
put any items in your shopping cart.</div>
        </div>
      {% else %}
        {% for item in cart().items %}
          {% if item.isVisible() %}
            {% snippet "cart-item" %}
```

```

        {% endif %}
    {% endfor %}
{% endif %}
</div>

<div class="pb-2.5 md:pb-3.5 lg:pb-6 w-full text-sm text-left">
    <form action="{{ route('cart.discount') }}" method="post">
        {{ csrf_field() }}
        <div class="flex flex-col space-y-2">
            <label for="cart-panel-discount-input" class="copy copy-text text-sm">Discount Code?</label>
            <div class="flex space-x-2">
                <input id="cart-panel-discount-input"
                    type="text"
                    class="border border-gray-300 w-full px-2.5 py-2"
                    placeholder="Enter Code"
                    name="code"
                    autocapitalize="off"
                    autocomplete="off"
                    spellcheck="false" />
                <button type="submit" class="px-5 py-3 text-xs font-medium transition border border-[color:#000000] text-[color:#000000] hover:text-[color:#030712]">Apply</button>
            </div>
        </div>
    </form>
</div>

<div class="w-full [&*]:py-1 text-sm">
    <div class="flex">
        <div class="font-medium">Subtotal</div>
        <div class="flex-1 text-right">{{ cart().subtotal }}</div>
    </div>
    {% for discount in cart().discounts %}
        <div class="flex">
            <div class="font-medium">
                {% if discount.code is not empty %}
                    Discount ({{ discount.code }})
                <form action="{{ route('cart.discount') }}" method="post">
                    {{ method_field('delete') }}
                    {{ csrf_field() }}

```

```

        <button type="submit"
            name="key" value="{{ discount.key }}"
            class="mt-3 inline-block text-xs font-semibold text-
gray-600 transition hover:text-gray-900 lg:mt-0">
            × remove
        </button>
    </form>
    {% else %}
        {{ discount.name }}
    {% endif %}
</div>
<div class="flex-1 text-right">- {{ discount.price.inc }}</div>
</div>
{% endfor %}
{% if cart().shippable %}
    <div class="flex">
        <div class="font-medium">Shipping</div>
        <div class="flex-1 text-right">
            {% if shipping %}
                {{ shipping.inc }}
            {% else %}
                <span class="text-xs text-gray-600">Calculated at next step</span>
            {% endif %}
        </div>
    </div>
{% endif %}
{% for surcharge in cart().surcharges %}
    <div class="flex">
        <div class="font-medium">{{ surcharge.name }}</div>
        <div class="flex-1 text-right">{{ surcharge.total_price }}</div>
    </div>
{% endfor %}
{% for voucher in cart().vouchers %}
    <div class="flex">
        <div class="font-medium">
            {% if voucher.code is not empty %}
                Voucher ({{ voucher.code }})
            {% else %}
                Voucher
            {% endif %}

```

```

        <form action="{{ route('cart.discount') }}" method="post">
            {{ method_field("delete") }}
            {{ csrf_field() }}
            <button type="submit"
                name="key" value="{{ voucher.key }}"
                class="mt-3 inline-block text-xs font-semibold text-gray-
600 transition hover:text-gray-900 lg:mt-0">
                × remove
            </button>
        </form>
    </div>
    <div class="flex-1 text-right">- {{ voucher.price.inc }}</div>
</div>
{% endfor %}
<div class="flex text-base">
    <div class="flex flex-wrap items-center text-left">
        <span class="block w-full font-medium">Total</span>
        {% if cart().total.hasTax() %}
            <span class="block text-xs text-gray-600">Including taxes of {{
cart().total.tax }}</span>
        {% endif %}
    </div>
    <div class="flex-1 text-right text-lg font-medium">{{ cart().total.inc
}}</div>
</div>
</div>

<div class="sticky bottom-0 pt-2.5 md:pt-3.5 lg:pt-6 pb-1 lg:pb-2.5">
    <a href="{{ route('checkout') }}" class="px-5 py-3 font-medium transition bg-
[color:#000000] text-[color:#FFFFFF] hover:bg-[color:#030712] cta-button inline-block mx-auto
text-sm text-center w-full">Checkout Securely</a>
</div>
</div>
</div>

{% js %}
    <div id="cart-panel-cover" class="cart-cover cart-cover--panel fixed inset-0 z-[998]"
style="background-color: rgba(0, 0, 0, 0.3)"></div>
    <script>
        !(() => {

```

```

let timeout, animate = true;

document.querySelector('body').classList.add('reduced-animation');

const open = () => {
  clearTimeout(timeout);

  if (animate) {
    document.querySelector('body').classList.remove('reduced-animation');
  }

  document.querySelector('body').classList.add('cart-panel--open');

  timeout = setTimeout(() =>
document.querySelector('body').classList.add('reduced-animation'), 500);
  }

const close = () => {
  clearTimeout(timeout);

  if (animate) {
    document.querySelector('body').classList.remove('reduced-animation');
  }

  document.querySelector('body').classList.remove('cart-panel--open');

  timeout = setTimeout(() =>
document.querySelector('body').classList.add('reduced-animation'), 500);
  }

AeroEvents.on('product.added-to-cart', function (data) {
  open();

  const parser = new DOMParser();
  const doc = parser.parseFromString(data.cart_panel, 'text/html');

  document.getElementById('cart-panel').innerHTML = doc.querySelector('#cart-
panel').innerHTML;
});

```

```

        window.addEventListener('DOMContentLoaded', () => {
            document.querySelector('#cart-panel-cover').addEventListener('click', close);

            document.querySelectorAll('.header-link--cart').forEach((el) =>
el.addEventListener('click', (e) => {
                e.preventDefault();

                open();
            }));
        });
    </script>
{% endjs %}

{% css %}
    <style>
        .cart-panel { transform: translateY(100%); @media (min-width: 768px) { transform:
translateX(100%); } }
        .cart-cover { transform: translateX(100%); opacity: 0; }
        .cart-panel--open .cart-panel { transform: translateY(0); @media (min-width: 768px) {
transform: translateX(0); } }
        .cart-panel--open .cart-cover--panel { transform: translateX(0); opacity: 1; }
        body:not(.reduced-animation) .cart-panel { transition: transform 300ms; }
        body:not(.reduced-animation) .cart-cover { transition: opacity 300ms; }
    </style>
{% endcss %}

```

This file uses the `snippets/cart-item.twig` snippet:

```

<div class="flex flex-wrap py-1.5 first:pt-0 last:pb-0 text-left">
    <div class="flex items-start py-2 pr-2 text-xs lg:py-4 max-w-[80px]">
        {% if item.url %}<a href="{{ item.url }}" aria-label="{{ item.name }}">{% endif %}
            {% set image_width = 320 %}
            {% set image_height = 320 %}
            {% if item.image %}
                
    {% else %}
        
    {% endif %}
    {% if item.url %}</a>{% endif %}
</div>
<div class="relative flex flex-1 flex-wrap">
    <div class="w-full py-2 lg:w-4/6 lg:flex-1 lg:p-4">
        <div class="flex">
            {% if item.url %}<a href="{{ item.url }}" class="inline-block text-sm no-
underline lg:hover:underline">{% endif %}
                {{ item.name }}
            {% if item.url %}</a>{% endif %}
        </div>
        {% if item.options|length %}
            <div class="mt-2 text-xs text-gray-600">
                {% for option in item.options %}
                    <div class="mt-px">
                        {% set value = option.display_value|default(option.value) %}
                        {% if value is empty %}
                            <span class="font-medium">{{ option.name }}</span>
                        {% else %}
                            <span class="font-medium">{{ option.name }}:</span> {{ value
}}

                        {% endif %}
                        {% if option.hasPrice() %}
                            ({{ option.price.inc }})
                        {% endif %}
                        {% if option.hasFiles() %}
                            [{{ option.fileCount() }} attached files]
                        {% endif %}
                    </div>
                {% endfor %}
            </div>
        {% endif %}
        {% if item.hasSubItems() %}

```

```

<div class="text-grey-600 mt-2 text-sm">
  {% for sub_item in item.subItems %}
    {% if sub_item.isHidden() %}
      <div class="{% if errors.has(sub_item.id) or sub_item.hasError()
%}text-red-800{% endif %}">
        {{ sub_item.quantity }} x {{ sub_item.name }}
        {% if not sub_item.isFree() %}
          &ndash; {{ sub_item.subtotal.inc }}
        {% endif %}
        {% if errors.has(sub_item.id) or sub_item.hasError() %}
          <span class="mt-3 block w-full text-sm font-medium text-
red-800" role="alert">{{ errors.first(sub_item.id)|default(sub_item.error) }}</span>
        {% endif %}
        {% if sub_item.canBeRemoved() %}
          <form class="ml-1 inline-block" action="{ {
route("cart.update") }}" method="post">
            {{ method_field("put") }}
            {{ csrf_field() }}
            <input type="hidden" name="cart_item_id" value="{ {
sub_item.id }}">
            (<button type="submit" name="quantity" value="0"
class="inline-block text-xs font-semibold text-gray-600 transition hover:text-gray-900">x
remove</button>)

            </form>
          {% endif %}
        </div>
      {% endif %}
    {% endfor %}
  </div>
{% endif %}
{% for file in item.files %}
  {% if file.hasExtension(["jpg", "jpeg", "png"]) %}
    
    {% endif %}
  {% endfor %}
{% if errors.has(item.id) or item.hasError() %}

```

```

        <span class="mt-3 block w-full text-sm font-medium text-red-800"
role="alert">{{ errors.first(item.id)|default(item.error) }}</span>
        {% endif %}
    </div>
    <div class="flex w-full flex-wrap items-center justify-end py-2 lg:w-2/6 lg:flex-col
lg:p-4 lg:pr-0">
        <div class="flex-1 order-1 w-1/2 text-right lg:order-none lg:w-full">
            {% if not item.isFree() %}
                <div class="block text-sm {% if item.isReduced() %}text-red-700{% endif
%}">

                    <span class="price">{{ item.subtotal.inc }}</span>
                </div>
                {% if item.isReduced() %}
                    <div class="block text-xs line-through">
                        <span class="price">{{ item.fullSubtotal.inc }}</span>
                    </div>
                {% endif %}
                {% if item.quantity > 1 %}
                    {% set unit_price %}
                        <span class="price">{{ item.price.inc }}</span>
                    {% endset %}
                    <div class="block text-xs text-gray-600">{{ unit_price }} each</div>
                {% endif %}
            {% endif %}
        </div>
        <div class="w-1/2 lg:order-1 lg:mt-2 lg:w-full lg:text-right">
            {% set quantity %}
                <form action="{{ route('cart.update') }}" method="post" class="inline-flex
items-center">

                    {{ method_field('put') }}
                    {{ csrf_field() }}
                    <input type="hidden" name="cart_item_id" value="{{ item.id }}">
                    {% if item.canBeChanged() %}
                        <select name="quantity"
                            class="ml-2 font-medium transition border border-
[color:#000000] text-[color:#000000] hover:text-[color:#030712] py-1.5 pl-1.5 text-xs"
                            onchange="this.form.submit()">
                            {% if item.canBeRemoved() %}
                                <option value="0">0</option>
                            {% endif %}

```

```

        {% for qty in range(item.quantity > 2 ? item.quantity - 1 : 1,
item.quantity + 5) %}

            <option value="{{ qty }}" {% if qty == item.quantity
%}selected{% endif %}>{{ qty }}</option>

            {% endfor %}

        </select>

        {% else %}

            <span class="ml-1">{{ item.quantity }}</span>

        {% endif %}

        {% if item.canBeRemoved() %}

            <button type="submit" class="button button--icon ml-2 -mr-1.5
hidden p-1 sm:flex"

                name="quantity" value="0" aria-label="Remove item">

                <svg class="h-5 w-5" width="24" height="24" fill="none"
viewBox="0 0 24 24">

                    <path stroke="currentColor" stroke-linecap="round" stroke-
linejoin="round" stroke-width="1.5" d="M6.75 7.75L7.59115 17.4233C7.68102 18.4568 8.54622
19.25 9.58363 19.25H14.4164C15.4538 19.25 16.319 18.4568 16.4088 17.4233L17.25 7.75" />

                    <path stroke="currentColor" stroke-linecap="round" stroke-
linejoin="round" stroke-width="1.5" d="M9.75 7.5V6.75C9.75 5.64543 10.6454 4.75 11.75
4.75H12.25C13.3546 4.75 14.25 5.64543 14.25 6.75V7.5" />

                    <path stroke="currentColor" stroke-linecap="round" stroke-
linejoin="round" stroke-width="1.5" d="M5 7.75H19" />

                </svg>

            </button>

        {% endif %}

    </form>

    {% endset %}

    <div class="flex items-center whitespace-nowrap text-xs lg:justify-end">

        <span class="font-medium text-gray-600">Quantity:</span>&nbsp;{{ quantity

    }}

    </div>

</div>

</div>

</div>

```

Then we must extend the `\Aero\Store\Http\Responses\CartItemAdd` response builder in order to update the cart panel:

```
\Aero\Store\Http\Responses\CartItemAdd::extend(function
(\Aero\Store\Http\Responses\CartItemAdd $action, \Closure $next) {
    $action->setData('cart_panel', (string) view('layouts.partials.cart_panel'));
});
```

The `cart_panel` is retrieved by our `product.added-to-cart` listener and used to effectively re-render the cart panel to use the new cart data.

Now we can include the cart panel in `header.twig`:

```
{% include "layouts/partials/cart-panel.twig" %}
```

Finally, if you want the cart panel to open when the cart button (typically an anchor tag with `href={{ route('cart') }}`) is clicked (rather than redirecting to the cart page), then add the `header-link--cart` class to it.