Template Guide

Index

Custobar

Overview

Custobar is a multi-channel marketing tool with full email marketing capabilities.

To create an email template that works with Custobar is effortless. You just create a template as you would otherwise do and decorate it with Custobar specific classes.

In Custobar the templates are modular, meaning that an email template can contain zero or more modules. The modules within a template can be reordered, removed and new modules can be added.

Get started fast

The fastest way to generate HTML mark-up template and few basic modules is to use the Email Template Designer integrated on Custobar. This editor creates a basic template and modules for your further modifications.

You may also use any HTML markup as a basis of your template. If you'd like you can use existing templates like Cerberus[1].

In addition to using ready-made, battle-tested templates and modules for the basis of your new email templates, be sure to verify that they render properly using e.g. Litmus[2].

Best Regards,
Team Custobar

[1] tedgoas.github.io/cerberus

[2] litmus.com

Custobar

Anatomy of an Custobar Email Template

To make an email template editable within Custobar, you'll need to do three things.

1. Add a special class `cb-module-wrapper` to a wrapping element, which hosts the modules.
2. Add content placeholder `{{ content|safe }}`.
3. Add editor embedding scripts definition `{{ editor_scripts|safe }}`.

   This string will be replaced with code that makes the templates editable. During sending of the mail, the embed will be replaced with an empty string.

You may add initial modules for the template, by including them directly within the wrapper element. Use 'include' with the string made from your Custobar username, "__" and module number you wish to include.

Note! Do not put any other content within the wrapper element than the `{% if ... endif %}` directive.

```
<!DOCTYPE html>

<html>
  <!-- ... -->
  <body>
    <div class="wrapper">
      <div class="cb-module-wrapper">


          {% if content %}
             {{ content|safe }}
          {% else %}

          <!-- Add here initial modules for the template -->
          {% include “username__5" %}

          {% endif %}


      </div>
    </div>


    {{ editor_scripts|safe }}


  </body>
</html>
```

Custobar

Anatomy of an Custobar Banner Template

Banner template is similar to an email template.

In minimum, it must have an body element and a wrapper element with class `cb-module-wrapper` .

The banners are rendered directly to hosting web page, without enclosing them into an `iframe` element.

If you'd like to match styles in the editor with styles on the actual site, you may add `html` and `head` elements with links to stylesheets.

```
<body>
  <div class="cb-module-wrapper">
    {% if content %}
      {{ content|safe }}
    {% else %}
      <!-- Add here initial modules for the template -->

      <div class="cb-module">
          My example module 1
      </div>
    {% endif %}
  </div>
  {{ editor_scripts|safe }}
</body>
```

Custobar

Anatomy of an Custobar Content Module

Custobar email content modules are simple DOM tree fragments as shown on the right.

The root level element must have a class `cb-module` and all the editable areas must have class `cb-module-wysiwyg`. The editable areas must be `div` elements.

To make an area editable in the detail pane and you may add class `cb-module-inline`, and attributes `data-editor-title` and `data-editor-widget`.

The editor title is the label of the field in the detail pane and widget is the input element. Valid input elements are `textarea` and `text-line`.

Images can be dragged onto `img` elements with class `cb-module-image`.

Note! `cb-module-wysiwyg` class must be set to a `div` element.

```
<div class="cb-module">
  <h1>A fixed title</h1>
  <p>A block of text, which cannot be edited in Custobar.</p>

  <img src="..." class="cb-module-image" />

  <h2>A fixed subtitle</h2>
  <div class="cb-module-wysiwyg">
    <p>A section which can be edited using the Custobar WYSIWYG
       editor.</p>
  </div>


  <h2>A fixed subtitle</h2>
  <div class="cb-module-wysiwyg">
    <p>Another editable section.</p>
  </div>

  <address class="cb-module-inline" data-editor-title="Address"
      data-editor-widget="text-line">Main Street 1</address>
</div>
```

Custobar

Anatomy of an Custobar Product Module

A product stored in Custobar can be dragged onto the email module, if it has one or more `cb-product` classes.

When dropped, the editor will replace

- `cb-module-image` with the product image
- href attribute of element with class `cb-product-link` with the product url.
- Text content of the elements `cb-product-title`, `cb-product-description`, `cb-product-price` and `cb-product-strike-price` with their respective contents.

You can make the fields, such as cb-product-title editable if you like, by adding the classes `cb-module-wysiwyg` or `cb-module-inline`.

Note! The white-space inside editable elements is significant, and is not collapsed into one space in the content editor like web browsers do it.

```
<div class="cb-module">
  <!- Example module containing two products, side by side -->

  <div class="cb-product">
    <img src="" class="cb-module-image" />
    <a href="#"
      class="cb-product-link cb-product-title">Title
      Placeholder</a>
  </div>

  <div class="cb-product">
    <a href="#" class="cb-product-link">
      <img src="..." class="cb-module-image" />
    </a>
    <span class="cb-product-title" data-editor-title="Title"
        data-editor-widget="text-line">Title Placeholder</span>
    <span class="cb-product-price">10,00 €</span>
  </div>
</div>
```

Custobar

Call-to-Action Module

A call-to-action module is a content module, with 1 to 5 buttons, each with its own unique action and redirection link. The buttons are edited with email content editor. To create a call-to-action module, decorate a normal module as follows.

- Decorate the element wrapping buttons with class `cb-cta-module-btns`, and add two attributes to it, shown on right.
  - `data-cb-cta-mail-width` is the pixel width of the entire message
  - `data-cb-button-widths` is a JSON array of five button widths, where the first width is for a single button and the last a width for each of the five buttons.
- Add a single element as the direct child of `cb-cta-module-btns`, and create your button layout with it. Decorate one of the descendant elements with class `cb-cta-module-btn`. This will be the button that is managed by Custoboar.

Custobar manages the buttons, so that it clones the first direct child element of `cb-cta-module-btns` and updates the `data-*` attributes of the nested descendant, which has the class `cb-cta-module-btn`.

```
<div class="cb-module">
  <!-- Other module content, typically headline etc -->

  <div class="cb-cta-module-btns"
    data-cb-cta-mail-width="700"
    data-cb-cta-button-widths="[300, 225, 175, 125, 100]">
    <div>
      <!- Other button content, e.g. decorations -->
      <a
        class="cb-cta-module-btn my-class"
        data-cb-cta-action="MAIL_SUBSCRIBE"
        data-cb-cta-value="my-newsletter"
        data-cb-cta-redirect="https://www.example.org"
        href="#">Button label</a>
    </div>
  </div>
</div>
```

NOTE

Provide defaults for data-cb-cta-* fields.

The possible options for action are, MAIL_SUBSCRIBE, MAIL_UNSUBSCRIBE, TAG and UNTAG.

Value is the mailing list or the tag name, and the redirect the URL address where the customer should be redirected after the button click has been handled.

Custobar

Custom Product Fields

Sometimes it is useful to include values from product data to the template.

For example, if you have a tax rate, which depends on the product type, and you want to add it to the template, you may do so by defining the name of the product field into `data-product-field` attribute as shown on the right.

If the product does not have the defined field, or its value is `null` a default value can be defined to field `data-product-field-default`.

The contents of the element with the attribute is erased, and replaced with the string value.

```
<div class="cb-product">
  <!-- ... -->

  <div data-product-field="COMPANY_ID__tax_rate"
      data-product-field-default="24 %">TAX</div>

  <!-- ... -->
</div>
```

Custobar

UTM Tags

If you'd like Custobar to add utm tags automatically to links, add class `data-utm-campaign` with value of your choosing to body element. Custobar will then add the following parameters to links in the email:

- utm_medium = email
- utm_source = custobar
- utm_campaign = YOUR_CAMPAIGN_ID

If you'd like to make `utm_campaign` parameter unique per campaign, you may add a special variable `{{ campaign_slug }}` as the value of `data-utm-campaign` attribute.

Additionally, if you like to have the optional `utm_content` parameter in the link, you may define `data-utm-content` attribute in any of the links parent elements' with a hard-coded value.

```
<!DOCTYPE html>

<html>
  <!-- ... -->
  <body data-utm-campaign="YOUR_CAMPAIGN_ID">
    <div class="wrapper">
      <div class="cb-module-wrapper">
        {% if content %}
          {{ content|safe }}
        {% else %}
          <!-- Add here initial modules for the template -->

          <div class="cb-module" data-utm-content="hero">
              My example module 1
          </div>
        {% endif %}
      </div>
    </div>
    {{ editor_scripts|safe }}
  </body>
</html>
```

Custobar

Unsubscribe and View as Web Page Links

Custobar creates always a web page for each email being sent, apart from transactional mails, which are tailored for individual user and often contain confidential data.

To link to this automatically generated page you may embed a special variable `--weblink--`, which is replaced with the actual url.

If you'd like Custobar to manage the unsubscriptions and provide an unsubscribe page, you may also add a link to `--unsubscribe_link--`.

```
<div class="cb-module">
  <div>
    An example module containing both the view as web
    page link and the unsubscribe link.
  </div>
  <a href="--weblink--">View as Web Page</a>
  <a href="--unsubscribe_link--">Unsubscribe here</a>
</div>
```

Custobar

Currency

Custobar templates support defining of currency on per template basis.

The default currency, if not defined, is Euro.

To define an alternative currency for the template, you'll need to define the following variables to the `html` element:

`data-currency-symbol` - The currency symbol, e.g. " ¥ ", "kr" or "CHF".

`data-currency-symbol-position` – The position of the symbol in relation to the value. Either "prefix" or "postfix". Defaults to "postfix".

`data-currency-symbol-spaced` – "true" or "false" depending whether to add a space between the symbol and the value, defaults to "true".

`data-price-fraction-digits` – Number of digits to show of the price. To drop the digits from the price, use 0. Defaults to 2.

```
<!DOCTYPE html>

<html data-currency-symbol="¥"
      data-currency-symbol-position="prefix"
      data-currency-symbol-spaced="false"
      data-price-fraction-digits="0">
  <!-- ... -->
</html>
```

Custobar

Enabling Custom Coloring of Elements

By default the users cannot change foreground and background colors of module elements (apart from wysiwyg areas).

To override this default behaviour, you may define one or more elements, whose foreground and background can be altered.

The colors that can be chosen are theme colors, defined in the settings and web safe colors.

To enable picking of colors, you'll need to add a special class `cb-mutable-bg-color`, `cb-mutable-fg-color` or both for an element.

In addition to these classes, you'll need to define a name for the color being edited. This is done via custom attributes `data-bg-color-name` and `data-fg-color-name`.

If you define the same name to multiple elements, all the elements with same name, receive the same color when picked.

On the right, you see an example of module containing a button, whose background and text color can be modified.

```
<div class="cb-module">
  <div>
    A lead paragraph for a button.
  </div>
  <a href="..." class="cb-mutable-bg-color"
      data-bg-color-name="Button Background Color">
    <span class="cb-mutable-fg-color"
        data-fg-color-name="Button Text Color">
      Button Text
    </span>
  </a>
</div>
```

Custobar

Background images

To drag and drop of background images, add a special class `cb-background-image` to an element.

By doing so, you'll allow user to drop image onto the module.

Custobar replaces background attribute of the element, where the aforementioned class. Additionally, to make the background image to appear in Outlook, you may add the comments on the right as direct children for the element.

The src attribute of any VML code element will be replaced also with the image url.

```
<div class="cb-module">
  <div class="cb-background-image">
    <!--[if gte mso 9]>
      <v:rect xmlns:v="urn:schemas-microsoft-com:vml"
          fill="true" stroke="false"
          style="mso-width percent:1000;">
        <v:fill type="tile" color="#7bceeb" />
        <v:textbox style="mso-fit-shape-to-text:true"
            inset="0,0,0,0">
      <![endif]-->

    <div>
      Content.
    </div>

    <!--[if gte mso 9]></v:textbox></v:rect><![endif]-->
  </div>
</div>
```

Custobar

Other Resources

- Embedding Custom Fonts
  https://litmus.com/blog/the-ultimate-guide-to-web-fonts

Custobar