

Using the Jinja Templating Language

02/16/2026 4:26 pm EST

Send highly personalized emails to your customers with personalized elements such as customer attributes (name, email address) or product attributes (product category, name). Bluecore personalizes emails with Jinja templating language to add additional customization to content.

 Bluecore supports everything that's available in the [Jinja](#) templating language. We've also added additional custom filtering for particular use cases.

Please note that this is an *advanced* article requiring a technical skill set.

Looking for examples of how to use Jinja in subject lines? Check out [this article](#) for some of our suggested use cases.

Getting started

Before adding customer attributes to your email campaigns, be sure to review the data that's available within your account through the [Customer 360](#) page. You can see the product attributes available within your account through the [Product Catalog](#) page. If there is a specific attribute you'd like to use, such as customer first name, last name; please be sure this is included in your Bluecore integration.

 Learn more about how to use Jinja in tracking parameters [here](#).

 Using invalid Jinja in email templates can cause the template to not render in the summary tab of email campaign building. When adding Jinja to your templates, make sure the template correctly renders before setting the campaign live. If the template preview is not rendering in the campaign summary page, link tracking will not function. Even if proofs render and the campaign sends as expected, link tracking is tied to the template preview. For the entire time the template is invalid, link tracking will be blocked and unrecoverable.

Customer attributes

Dynamic Attribute	Description
<code>{{customer.email}}</code>	Display customer's email address
<code>{{customer.attribute_name}}</code>	Use this attribute to display any additional customer attributes that may be available within Bluecore, such as first name, gender, etc.

Campaign attributes

Dynamic Attribute	Description
<code>{{campaign_name}}</code>	Campaign Name
<code>{{nqe.subaction}}</code>	Campaign ID
<code>©{{nqe.created pretty_time('%Y')}}}</code>	©2021

Dynamic product attributes

For the following attributes, be sure to adjust the below templated Jinja syntax to match the blocks, products, etc. within

your email. For example, the dynamic attribute of `{{BLOCKNAME.products.0.url}}`, the *BLOCKNAME* should be replaced with the actual name of the block in the email.

The below attributes can be configured to display the most common value (mode) of the products or the value of the first product.

This is only a sampling of the attributes that are available for your account. Please see the [Product Details Page](#) within your account for the full list.

Dynamic Attribute	Example
<code>{{BLOCKNAME.products mode('category')}} OR {{BLOCKNAME.products.0.category}}</code>	Shoes
<code>{{BLOCKNAME.products mode('brand')}} OR {{BLOCKNAME.products.0.brand}}</code>	Nike
<code>{{BLOCKNAME.products mode('name')}} OR {{BLOCKNAME.products.0.name}}</code>	Air Max
<code>{{BLOCKNAME.products.0.url}}</code>	www.URLHERE.com

Price attributes

We recommend using the Dynamic Product Blocks in VTE. If you must use an advanced custom code block to display product information, you can use this syntax.

Filter	Example Syntax	Example
Filter	Example Syntax	Example
money_format_with_commas	<code>{{product.price money_format_with_commas(currency="£", separator=",", decimal=".", use_cents=False)}}</code>	Input: 242000Output: £242,000
Money_format_with_locale_and_currency	<code>{{product.price money_format_with_locale_and_currency("en-US_USD_\$\$,##0.##")}}</code>	Input: 1229.6789Output: \$1,229.68 <i>Values inside the () can be customized</i>
Advanced price conditional example	<code>{% if product.price float > 5 %} {{product.price money_format}}{% endif %}</code>	Product prices are strings by default. Run this filter to do numerical comparisons. Please note that negatives are not supported at the beginning of a comparison.
Advanced price conditional example	<code>{% if product.on_sale %} Now: {{product.price money_format}} Original: {{product.original_price money_format}} {% else %} {{product.price money_format}} {% endif %}</code>	Use this if you want to display the price differently if the product is on sale.

Advanced Jinja filters

Filter	Example Syntax	Example
Upper Case	<code>{{BLOCKNAME.products.0.name upper}}</code>	Input: "Air Max"Output: "AIR MAX"
Lower Case	<code>{{BLOCKNAME.products.0.name lower}}</code>	Input: "Air Max"Output: "air max"
Capitalize	<code>{{BLOCKNAME.products.0.name capitalize}}</code>	Input: "air max"Output: "Air max"
Title	<code>{{BLOCKNAME.products.0.name title}}</code>	Input: "air max"Output: "Air Max"
Replace	<code>{{BLOCKNAME.products.0.name replace("-", " ")}}</code>	Input: air-maxOutput: air max
Split	<code>{{BLOCKNAME.products.0.name split("-")}}</code>	Input: air-maxOutput: "air"
Trim	<code>{{BLOCKNAME.products.0.name trim}}</code>	Input: " air max "Output: "air max"
Slugify	<code>{{BLOCKNAME.products.0.name slugify}}</code>	Input: "Air Max"Output: "air-max"
Unhyphenate	<code>{{BLOCKNAME.products.0.name unhyphenate}}</code>	Input: "Air-Max"Output: "Air Max"
pretty_time	<code>{{nqe.created}}</code> <code>{{nqe.created pretty_time("%Y%m%d")}}</code> <code>{{nqe.created pretty_time("%m/%d/%Y %H:%M")}}</code>	Formats time in the specified inputs.
add_time	Hurry, this offer ends <code>{{nqe.created add_time(3,'days') pretty_time("%m/%d")}}</code>	Add additional time to a given time. For example, this offer ends 2/18 .
to_datetime	<code>{{product.datetime_str to_datetime()}}</code> <code>{{product.datetime_str to_datetime("%Y%m%d")}}</code> <code>{{product.datetime_str to_datetime("%Y%m%d") pretty_time("%Y")}}</code>	Turns a string into a datetime object. This can be combined with pretty_time.
hasattr	<code>{% if product hasattr('description') %}</code> <code>{{product.description}}{% endif %}</code>	Returns a boolean value based on if an object has a given attribute.
product_any	<code>{% if products product_any('ineligible_for_coupon', [True]) %}<!-- show specific legal copy -->{% endif %}</code>	Returns a boolean value based on if any product in a set has a given attribute set at a specific value.