Jinja

08/01/2025 2:55 pm EDT

Using the Jinja Templating Language

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. For a more basic understanding of dynamic attributes in Bluecore, click here.

Looking for examples of how to use Jinja in subject lines? Check outthis 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 parametershere.

⚠ Using invalid Jinja in email templates can cause the template to not render in thesummary 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
`#{{customer.email}}`	Display customer's email address
`#{{customer.attribute_name}}`	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
`#{{campaign_name}}`	Campaign Name
`#{{nqe.subaction}}`	Campaign ID
`©#{{nqe.created|pretty_time('%Y')}}`	©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 theProduct Details Page within your account for the full list.

Dynamic Attribute	Example
`#{{BLOCKNAME.products|mode('category')}}` or `#{{BLOCKNAME.products.0.category}}`	Shoes
`#{{BLOCKNAME.products|mode('brand')}}` or `#{{BLOCKNAME.products.0.brand}}`	Nike
`#{{BLOCKNAME.products|mode(‘name’)}}` or `#{{BLOCKNAME.products.0.name}}`	Air Max
`#{{BLOCKNAME.products.0.url}}`	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	`# {{product.price|money_format_with_commas(currency="£", separator=",", decimal=".", use_cents=False)}}`	Input: 242000Output: £242,000
Money_format_with_locale_and_currency	`# {{product.price|money_format_with_locale_and_currency("en-US_USD_$#,##0.##")}}`	Input: 1229.6789Output: $1,229.68 *Values inside the () can be customized*
Advanced price conditional example	`{% if product.price|float > 5 %} # {{product.price|money_format}}{% endif %}`	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	`{% if product.on_sale %}Now: # {{product.price|money_format}} Original: # {{product.original_price|money_format}}{% else %}# {{product.price|money_format}}{% endif %}`	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	#{{BLOCKNAME.products.0.name|**upper**}}	Input: “Air Max”Output: “AIR MAX”
Lower Case	#{{BLOCKNAME.products.0.name|**lower**}}	Input: “Air Max”Output: “air max”
Capitalize	#{{BLOCKNAME.products.0.name|**capitalize**}}	Input: “air max”Output: “Air max”
Title	#{{BLOCKNAME.products.0.name|**title**}}	Input: “air max”Output: “Air Max”
Replace	#{{BLOCKNAME.products.0.name|**replace**("-", " ")}}	Input: air-maxOutput: air max
Split	#{{BLOCKNAME.products.0.name|**split**("-")}}	Input: air-maxOutput: “air”
Trim	#{{BLOCKNAME.products.0.name|**trim**}}	Input: “ air max "Output: “air max”
Slugify	#{{BLOCKNAME.products.0.name|**slugify**}}	Input: “Air Max”Output: “air-max”
Unhyphenate	#{{BLOCKNAME.products.0.name|**unhyphenate**}}	Input: “Air-Max”Output: “Air Max”
pretty_time	#{{nqe.created}} #{{nqe.created |**pretty_time**("%Y%m%d")}} #{{nqe.created |**pretty_time**("%m/%d/%Y %H:%M")}}	Formats time in the specified inputs.
add_time	Hurry, this offer ends #{{nqe.created|**add_time**(3,'days')|pretty_time('%m/%d')}}	Add additional time to a given time. For example, **this** offer ends **2/18**.
to_datetime	#{{product.datetime_str|**to_datetime**()}} #{{product.datetime_str |**to_datetime**('%Y%m%d')}} #{{product.datetime_str| **to_datetime**('%Y%m%d')|**pretty_time**('%Y')}}	Turns a string into a datetime object. This can be combined with pretty_time.
hasattr	{% **if** product|**hasattr**('description') %}#{{product.description}}{% endif %}	Returns a boolean value based on if an object has a given attribute.
product_any	{% **if** products|**product_any**('ineligible_for_coupon', [True]) %}<!-- show specific legal copy -->{% endif %}	Returns a boolean value based on if any product in a set has a given attribute set at a specific value.

Jinja Templating Language Examples

The below Jinja subject line examples are *only* examples from Bluecore campaign types. Please be sure to modify the below Jinja syntax before adding them to an email campaign.

ⓘ Each example of Jinja requires a specific set of data that must be available within Bluecore. You can see available data in your account through the Customer 360 and Product Catalog pages. Please be sure the appropriate data is being passed to Bluecore for the Jinja examples to properly display in emails.

Abandoned cart

- Re : #{{BLOCKNAME.products.0.name}} #{{BLOCKNAME.products.0.brand}}
- Your #{{BLOCKNAME.products.0.name|title }} Is Waiting For You
- {% if customer.first_name %}#{{customer.first_name}}, {% endif %}You Left Items in Your Cart

Abandoned product

- Your #{{BLOCKNAME.products.0.name}} is waiting!
- We like #{{BLOCKNAME.products.0.category|replace(':','') }}, too!
- You know what would go great with #{{BLOCKNAME.products.0.category|replace(':','') }}?

Abandoned search

- {% if customer.first_name %}#{{customer.first_name}}{% else %}Hey{% endif %}, you searched, we found ≣
- Did you forget your #{{ BLOCKNAME.products.0.name }}?
- Don't Forget About #{{BLOCKNAME.products.0.brand}}

New arrivals

- Have You Seen What’s New From #{{BLOCKNAME.products|mode('brand') }}?

Window shopping

- Your #{{BLOCKNAME.products.0.name}} is waiting!
- Our Best Selling #{{BLOCKNAME.products.0.category}}

Price decrease

- Good news! Our #{{BLOCKNAME.products.0.name}} has a new, lower price.
- Price drop on #{{BLOCKNAME.products.0.name}}
- Price Drop Alert: The #{{BLOCKNAME.products.0.name}} has a new, lower price.

Low inventory

- Better Hurry! The $#{{BLOCKNAME.products.0.price}} #{{block_0.products.0.name}} Is Going FAST
- {% if customer.first_name %}#{{customer.FIRST_NAME|title}}, c{% else %}C{% endif %}atch your faves!
- #{{BLOCKNAME.products.0.name}} is Almost Out of Stock

Best sellers

- #{{BLOCKNAME.products.0.name|title}} & Best Sellers Picked for You!

Setting a fallback

Including fallback language ensures that each customer will still receive something in the place of the dynamic attribute. If Bluecore is unable to pull in the specific customer attribute, the email will still go out with the fallback in place.

Example fallback: {% if customer.first_name %}Dear #{{customer.FIRST_NAME|title}}, {% else %}Dear Customer,{% endif %} catch your faves!

Testing

Be sure to test the email before sending. Review the code, test, and preview the email before hitting Save & Launch! Clickhere for more information on testing dynamic attributes before sending an email.