

ProductMeasurementUnitWidget module

Atoms

json-scheme (atom)

Renders a json data as an attribute into html tag.

Code sample

```
{% include atom('json-scheme', 'ProductMeasurementUnitWidget') with {
  data: {
    name: config.name,
    json: data.json
  }
} only %}
```

Molecules

measurement-quantity-selector (molecule)

Shows html tag element, which includes json data as a html attribute.

Code sample

```
{% include molecule('measurement-quantity-selector',
'ProductMeasurementUnitWidget') with {
  data: {
    json: data.json
  }
} only %}
```

ProductPackagingUnitWidget module

Atoms

json-scheme (atom)

Renders a json data as attribute into html tag.

Code sample

```
{% include atom('json-scheme', 'ProductPackagingUnitWidget') with {
  data: {
    name: config.name,
    json: data.json
  }
} only %}
```

```
    }  
  } only %}
```

Molecules

order-detail-product-packaging-unit (molecule)

Displays a list of order details as table which includes product sku, name, price, quantity, and item total.

Code sample

```
{% include molecule('order-detail-product-packaging-unit',  
'ProductPackagingUnitWidget') with {  
  data: {  
    concreteItem: item,  
    currencyIsoCode: data.currencyIsoCode,  
    concreteItemDefault: concreteItem  
  }  
} only %}
```

packaging-unit-cart (molecule)

Displays a product quantity and amount on checkout cart step.

Code sample

```
{% include molecule('packaging-unit-cart', 'ProductPackagingUnitWidget') with {  
  data: {  
    amount: data.cartItem.amount,  
    quantity: data.cartItem.quantity,  
    quantitySalesUnit: data.cartItem.quantitySalesUnit,  
    quantityConversion: data.cartItem.quantitySalesUnit.conversion,  
    quantityProductMeasurementUnit:  
data.cartItem.quantitySalesUnit.productMeasurementUnit,  
    amountSalesUnit: data.cartItem.amountSalesUnit,  
    amountConversion: data.cartItem.amountSalesUnit.conversion,  
    amountProductMeasurementUnit:  
data.cartItem.amountSalesUnit.productMeasurementUnit,  
    amountValue: data.cartItem.amountSalesUnit.value  
  }  
} only %}
```

packaging-unit-quantity-selector (molecule)

Shows html tag element, which includes json data as html attribute.

Code sample

```
{% include molecule('packaging-unit-quantity-selector',  
'ProductPackagingUnitWidget') with {  
  data: {  
    json: data.jsonScheme  
  }  
} only %}
```

summary-product-packaging-unit (molecule)

Displays a product summary price on checkout summary step.

Code sample

```
{% include molecule('summary-product-packaging-unit', 'ProductPackagingUnitWidget')  
with {  
  data: {  
    name: data.name,  
    quantity: data.quantity,  
    amount: data.amount,  
    price: data.price,  
    quantitySalesUnit: data.quantitySalesUnit,  
    amountSalesUnit: data.amountSalesUnit  
  }  
} only %}
```

ShopUi module

Atoms

badge (atom)

Creates a simple, colored label element.

Code sample

```
<span class="badge">Text</span>
```

box (atom)

Creates a simple parent container with the default border.

Code sample

```
<div class="box">Text</div>
```

button (atom)

Creates a simple, clickable button element.

Code sample

```
<button type="submit" class="button">Submit</button>
```

checkbox (atom)

Creates a simple checkbox element with the non-native design.

Code sample

```
{% include atom('checkbox') with {  
  attributes: {  
    name: 'name',  
    value: 1,  
    checked: true  
  },  
  data: {  
    label: 'label'  
  }  
} only %}
```

icon (atom)

Creates a simple icon element based on the sprite.

Code sample

```
{% include atom('icon') with {  
  data: {  
    name: 'name'  
  }  
} only %}
```

icon-sprite (atom)

Creates a sprite with the collection of svg icons that can be used via the icon component with a specific icon name.

Code sample

```
{% include atom('icon-sprite') only %}
```

input (atom)

Creates a simple input element - a control that allows the user to input data.

Code sample

```
<input name="name" type="type" class="input">
```

label (atom)

Creates a simple caption to describe a form element.

Code example

```
<label class="label">Text</label>
```

link (atom)

Creates a simple link element that enables visitors to click through to another web page (or another URL).

Code example

```
<a class="link" href="url">Text</a>
```

list (atom)

Creates a simple unordered list.

Code example

```
<ul class="list">
  <li class="list__item">Text</li>
  <li class="list__item">Text</li>
  <li class="list__item">Text</li>
</ul>
```

node (atom)

Creates a simple unordered list item element.

Code sample

```
{% include atom('node') with {
  config: {
    tag: 'li'
  }
}
```

```
    },
    data: {
      node: node
    }
  } only %}
```

radio (atom)

Creates a radio button element that allows the user to select one option from a range of options with the non-native design.

Code sample

```
{% include atom('radio') with {
  attributes: {
    name: 'name',
    checked: true
  }
} only %}
```

select (atom)

Creates a simple, single-selection drop-down element.

Code sample

```
{% include atom('select') with {
  data: {
    options: options
  },
  attributes: {
    name: 'name',
    disabled: false
  }
} only %}
```

table (atom)

Implements basic styles for tables, spacings, borders and so on.

Code sample

```
<table class="table">
  <thead></thead>
  <tbody></tbody>
  <tfoot></tfoot>
</table>
```

textarea (atom)

Implements basic styles for the textarea element, like spacing, borders and so on.

Code sample

```
<textarea class="textarea">Text</textarea>
```

thumbnail (atom)

Creates a simple image element.

Code sample

```
{% include atom('thumbnail') with {
  attributes: {
    alt: 'alt',
    src: src,
    title: 'title'
  }
} only %}
```

Molecules

action-single-click-enforcer (molecule)

Prohibits the double click event on a specific target (array of DOM-elements).

Code sample

```
{% include molecule('action-single-click-enforcer') with {
  attributes: {
    'target-selector': 'target-selector'
  }
} only %}
```

ajax-loader (molecule)

Displays a "loading" indicator image (spinner) while an AJAX request sent via the related ajax-provider is being performed.

Code sample

```
{% include molecule('ajax-loader') with {
  data: {
    label: 'label',
    showSpinnerOnLoad: false
  }
}
```

```
    attributes: {
      'provider-selector': 'provider-selector'
    }
  } only %}
```

ajax-provider (molecule)

Enables performing an ajax request to a specified url with some URL parameters and and response handling.

Code sample

```
{% include molecule('ajax-provider') with {
  attributes: {
    'fetch-on-load': false,
    'url': 'url',
    'method': 'method',
    'response-type': 'response-type'
  }
} only %}
```

ajax-renderer (molecule)

Renders HTML received from the related ajax-provider component into a specific DOM-element.

Code sample

```
{% include molecule('ajax-renderer') with {
  attributes: {
    'provider-selector': 'provider-selector',
    'render-if-response-is-empty': false,
    'target-selector': 'target-selector'
  }
} only %}
```

autocomplete-form (molecule)

Enables users to quickly find and select from a pre-populated list of values as they type, leveraging searching and filtering.

Code sample

```
{% include molecule('autocomplete-form') with {
  attributes: {
    'suggestion-url': 'suggestion-url',
    'selected-value-key': 'selected-value-key',
    'value-data-attribute': 'value-data-attribute',
    'item-selector': 'item-selector',
    'show-clean-button': false
  }
}
```



```
} only %}
```

breadcrumb (molecule)

A "breadcrumb" is a type of the secondary navigation scheme that reveals user's location on a website.

Code sample

```
{% include molecule('breadcrumb') with {
  data: {
    steps: [{
      label: 'label',
    },
    {
      label: 'label',
    }],
    startWithHome: true
  }
} only %}
```

breadcrumb-step (molecule)

A simple list of items that displays a hierarchical structure of breadcrumbs.

Code sample

```
{% embed molecule('breadcrumb-step') with {
  data: {
    itemProp: null,
    label: 'label',
    url: 'url',
    withChevron: false
  }
} only %}
```

card (molecule)

Used for a short display of a product card (article, post), which may contain a title, an image, a description, and a button.

Code sample

```
{% include molecule('card') with {
  data: {
    name: 'name',
    url: 'url',
    imageUrl: 'imageUrl',
    description: 'description'
  }
}
```

```
} only %}
```

cart-items-counter (molecule)

Displays the number of items in a product cart.

Code sample

```
{% include molecule('cart-items-counter') with {  
  data: {  
    label: 'label',  
    quantity: quantity  
  }  
} only %}
```

display-address (molecule)

Displays a customer shipping/billing address.

Code sample

```
{% include molecule('display-address') with {  
  data: {  
    address: 'address'  
  }  
} only %}
```

flash-message (molecule)

Displays a simple notification that informs users about important events that may have happened.

Code sample

```
{% include molecule('display-address') with {  
  data: {  
    action: 'action',  
    title: 'title',  
    text: 'text',  
    icon: null  
  }  
} only %}
```

form (molecule)

Wraps the symfony form to render its elements in a specific way; consists of a title, a list of form elements, the "submit" and the "cancel" buttons.

Code sample

```
{% include molecule('form') with {
  data: {
    form: form,
    enableStart: true,
    enableEnd: true,
    layout: {},
    options: {
      attr: {
        novalidate: 'novalidate'
      }
    },
    title: 'title'
  }
} only %}
```

form-input-default-value-disabler (molecule)

Disables form fields before submitting if they have the data-default-value attribute.

Code sample

```
{% include molecule('form-input-default-value-disabler') with {
  attributes: {
    'form-selector': 'form-selector',
    'input-selector': 'input-selector',
    'default-value-attribute': 'default-value-attribute'
  }
} only %}
```

jumbotron (molecule)

Displays a banner with an image and a short description.

Code sample

```
{% include molecule('jumbotron') with {
  data: {
    imageUrl: 'imageUrl',
    bgColor: 'bgColor',
    textColor: 'textColor'
  }
} only %}
```

logo (molecule)

Displays the Spryker default logo or, if provided, a custom logo. In either cases, the image is also a link to the homepage.

Code sample

```
{% include molecule('logo') only %}
```

menu (molecule)

Styles for the main navigation list.

Code sample

```
<ul class="menu">
  <li class="menu__item">Text</li>
  <li class="menu__item">Text</li>
  <li class="menu__item">Text</li>
</ul>
```

navigation-icon-list (molecule)

Creates a list of link or text items with icons inside.

Code sample

```
{% include molecule('navigation-icon-list') with {
  data: {
    nodes: nodes
  }
} only %}
```

navigation-list (molecule)

Creates a list of links or text items.

Code sample

```
{% include molecule('navigation-list') with {
  data: {
    nodes: nodes
  }
} only %}
```

navigation-multilevel (molecule)

Creates a list of navigation items, which makes it possible to render items recursively.

Code sample

```
{% include molecule('navigation-multilevel') with {
```

```
data: {
  nodes: nodes,
  menuInline: false,
  menuDropdown: false
}
} only %}
```

navigation-multilevel-node (molecule)

Creates a menu item: it can be a link, a text message or a list of links.

Code sample

```
{% include molecule('navigation-multilevel-node') with {
  data: {
    node: node
  }
} only %}
```

outside-widgets (molecule)

Create a container for widgets.

Code sample

```
{% include molecule('outside-widgets') only %}
```

pagination (molecule)

Creates a customizable pagination.

Code sample

```
{% include molecule('pagination') with {
  parameters: parameters,
  currentPage: currentPage,
  paginationPath: 'paginationPath',
  showAlwaysFirstAndLast: false,
  maxPage: maxPage,
  extremePagesLimit: extremePagesLimit,
  nearbyPagesLimit: nearbyPagesLimit,
  anchor: 'anchor'
} only %}
```

price (molecule)

Displays the product price with an/the icon and shows the original price in case of a discount.

Code sample

```
{% include molecule('price') with {
  data: {
    amount: 'amount',
    originalAmount: 'originalAmount'
  }
} only %}
```

price-mode (molecule)

Displays a simple label with icon.

Code sample

```
{% include molecule('price-mode') with {
  data: {
    priceMode: data.priceMode,
    showIcon: false
  }
} only %}
```

product-card (molecule)

Creates a block with product information: product labels, list of product group icons, product name, product rating, and product price.

Code sample

```
{% include molecule('product-card') with {
  data: {
    name: 'name',
    abstractId: abstractId,
    url: 'url',
    imageUrl: 'imageUrl',
    price: 'price',
    originalPrice: 'originalPrice'
  }
} only %}
```

product-list-item (molecule)

Creates a block with detailed product information: product labels, product image, list of product group icons, product name, product rating, product price, and a link to the product.

Code sample

```
{% include molecule('product-list-item') with {
  data: {
```

```
        name: 'name',
        abstractId: abstractId,
        url: 'url',
        imageUrl: 'imageUrl',
        price: 'price',
        originalPrice: 'originalPrice'
    }
} only %}
```

quantity-input (molecule)

Displays a product quantity field with the label and the optionally update button for this field.

Code sample

```
{% include molecule('quantity-input') with {
    data: {
        maxQuantity: maxQuantity,
        value: 'value',
        readOnly: false
    }
} only %}
```

quantity-selector (molecule)

Displays a drop-down with predefined values of product quantity.

Code sample

```
{% include molecule('quantity-selector') with {
    data: {
        maxQuantity: maxQuantity,
        step: step
    }
} only %}
```

script-loader (molecule)

Provides an async loading scripts to the DOM.

Code sample

```
{% include molecule('script-loader') with {
    attributes: {
        src: data.scriptSrc
    }
} only %}
```

search-form (molecule)

Creates a search input field, submit button to trigger search, and also consist suggest-search molecule which add search suggestions functionality to search input.

Code sample

```
{% include molecule('search-form') with {
  attributes: {
    'data-search-id': 'data-search-id'
  }
} only %}
```

simple-carousel (molecule)

Creates a carousel from an array of items, generates the left and right arrows as well as the optional dots control.

Code sapmle

```
{% include molecule('simple-carousel') with {
  data: {
    slides: slides,
    showDots: true
  },
  attributes: {
    'slides-to-show': slides-to-show,
    'slides-to-scroll': slides-to-scroll
  }
} only %}
```

suggest-search (molecule)

Implements search suggestions functionality for a search input, creates an empty container for search suggestion results, uses ajax-provider and ajax-loader.

Code sample

```
{% include molecule('suggest-search') with {
  attributes: {
    'suggestion-url': suggestion-url,
    'base-suggest-url': 'base-suggest-url',
    'input-selector': 'input-selector',
    'debounce-delay': debounce-delay,
    'throttle-delay': throttle-delay,
    'letters-trashold': letters-trashold
  }
} only %}
```


toggler-checkbox (molecule)

Creates a checkbox that adds/removes a class to/from some DOM-elements specified by the selector on check/uncheck.

Code sample

```
{% include molecule('toggler-checkbox') with {
  data: {
    type: 'type'
  },
  attributes: {
    'target-selector': 'target-selector',
    'class-to-toggle': 'class-to-toggle',
    checked: true
  }
} only %}
```

toggler-click (molecule)

Initiates functionality to handle clicks on the array of DOM-elements to toggle the specific class on another array of DOM-elements.

Code sample

```
{% include molecule('toggler-click') with {
  attributes: {
    'trigger-selector': 'trigger-selector',
    'target-selector': 'target-selector',
    'class-to-toggle': 'class-to-toggle'
  }
} only %}
```

toggler-hash (molecule)

Initiates functionality to add/remove class to/from some array of DOM-elements if browser address input field consist or doesn't consist same hash as in component.

Code sample

```
{% include molecule('toggler-hash') with {
  attributes: {
    'trigger-hash': 'trigger-hash',
    'target-selector': 'target-selector',
    'class-to-toggle': 'class-to-toggle',
    'add-class-when-hash-in-url': false
  }
} only %}
```

toggler-radio (molecule)

Initiates functionality to add a specific class to some element from an array of DOM-elements and remove it from the rest of the elements if a radio-button is selected.

Code sample

```
{% include molecule('toggler-radio') with {
  data: {
    type: 'type'
  }
} only %}
```

Organisms

footer (organism)

Creates a layout part at the bottom of the site that consists of the newsletter subscription form, payment and shipment lists of icons, contacts, and social icon links.

Code sample

```
{% include organism('footer') only %}
```

header (organism)

Creates a layout part at the top of the site that consists of top navigation, logo, search form and main site navigation.

Code sample

```
{% include organism('header') with {
  data: {
    showSearchForm: true,
    showNavigation: true
  }
} only %}
```

navigation-top (organism)

Creates a navigation with the list of items: price mode switcher, currency switcher, language switcher, Company name and Business Unit, mini-cart, quick order link, shopping list, company account link, link to customer profile, link to wishlist and login/logout link.

Code sample

```
{% include organism('navigation-top') with {
  data: {
    isInline: false,
    withSeparators: false
  }
} only %}
```

notification-area (organism)

Displays global messages at the top of the screen.

Code sample

```
{% include organism('notification-area') only %}
```

side-drawer (organism)

Displays side-drawer for mobile devices that duplicates top and main navigation.

Code sample

```
{% include organism('side-drawer') with {
  attributes: {
    'container-selector': 'container-selector',
    'trigger-selector': 'trigger-selector'
  }
} only %}
```

AgentWidget module

Molecules

customer-list (molecule)

Renders a list of customers with their first name, last name, and email.

Code sample

```
{% include molecule('customer-list', 'AgentWidget') with {
  data: {
    customers: customers
  }
} only %}
```

Organisms

agent-control-bar (molecule)

Displays a grid with the autocomplete form for user selection as well as a list of users with the logout link.

Code sample

```
{% include organism('agent-control-bar', 'AgentWidget') with {
  data: {
    agent: agent,
    customer: customer
  }
} only %}
```

AvailabilityWidget module

Molecules

availability-product (molecule)

Shows alert text message if product is not available.

Code sample

```
{% include molecule('availability-product', 'AvailabilityWidget') ignore missing
with {
  data: {
    idProductConcrete: data.product.idProductConcrete,
    isAvailable: data.product.available
  }
} only %}
```

CartNoteWidget module

Molecules

note-list (molecule)

Shows unordered list with optional label as first item and text note as second.

Code sample

```
{% include molecule('note-list', 'CartNoteWidget') with {
```

```
data: {
  label: data.label,
  note: data.note
}
} only %}
```

CartPage module

Molecules

cart-image (molecule)

Displays a product image or a text message that the image is not available if there is no image source.

Code sample

```
{% include molecule('cart-image', 'CartPage') with {
  data: {
    name: 'name',
    image: null
  }
} only %}
```

cart-item (molecule)

Displays detailed product information: product image, product name, product sku, drop-down with product variants, list of bundles, price, packaging unit information, quantity input, "remove" link, and a form with note.

Code sample

```
{% include molecule('cart-item', 'CartPage') with {
  data: {
    cart: cart,
    cartItem: cartItem,
    attributes: attributes
  }
} only %}
```

cart-item-summary (molecule)

Displays product summary information: price, price for different options, total, and subtotal.

Code sample

```
{% include molecule('cart-item-summary', 'CartPage') with {
```

```
    data: {
      unitPrice: 'unitPrice',
      subtotalPrice: 'subtotalPrice',
      cartItem: cartItem
    }
  } only %}
```

cart-item-variant-selector (molecule)

Displays a list of product variants as dropdowns.

Code sample

```
{% include molecule('cart-item-variant-selector', 'CartPage') with {
  data: {
    cartItem: cartItem,
    cartItemAttributes: cartItemAttributes
  }
} only %}
```

cart-quantity-input (molecule)

Displays the product quantity input that allows setting the maximum quantity and the "read-only" state for this input, as well as the "submit" button.

Code sample

```
{% include molecule('cart-quantity-input', 'CartPage') with {
  data: {
    cartItem: cartItem,
    readOnly: false
  }
} only %}
```

cart-summary (molecule)

Displays information about quantity of products in the shopping cart, cart subtotal, total, and grand total, as well as information about discounts and link to the checkout.

Code sample

```
{% include molecule('cart-summary', 'CartPage') with {
  data: {
    cart: cart,
    isQuoteValid: isQuoteValid
  }
} only %}
```

CartToShoppingListWidget module

Molecules

toggle-select-form (molecule)

The toggle-class for the specified DOM-elements on the target change event.

Code sample

```
{% include molecule('toggle-select-form', 'CartToShoppingListWidget') with {
  data: {
    attribute: attribute,
    selectAttributes: {
      'data-select-trigger': 'data-select-trigger',
      'target': 'target',
      'class-to-toggle': 'class-to-toggle'
    }
  }
} only %}
```

CatalogPage module

Molecules

active-filter (molecule)

Displays a selected filter as a link, filter gets disabled after clicking the link.

Code sample

```
{% include molecule('active-filter', 'CatalogPage') with {
  attributes: attributes,
  data: {
    filter: filter,
    value: value
  }
} only %}
```

active-filter-enumeration (molecule)

Displays a list of all the filters selected from one category in form of links.

Code sample

```
{% include molecule('active-filter-enumeration', 'CatalogPage') with {
  filter: filter,
```

```
isMultiSelect: false
} only %}
```

active-filter-multi-select (molecule)

Displays a list of all the filters selected from one category in form of links.

Code sample

```
{% include molecule('active-filter-multi-select', 'CatalogPage') with {
  data: {
    isMultiSelect: true
  }
} only %}
```

active-filter-price-range (molecule)

Displays a selected filter, which shows a price range.

Code sample

```
{% include molecule('active-filter-price-range', 'CatalogPage') with {
  data: {
    filter: filter,
    min: min,
    activeMin: activeMin,
    max: max,
    activeMax: activeMax
  }
} only %}
```

active-filter-rating (molecule)

Displays a selected filter, which shows a number of the rating chosen.

Code sample

```
{% include molecule('active-filter-rating', 'CatalogPage') with {
  data: {
    value: 'value'
  }
} only %}
```

active-filter-single-select (molecule)

Displays a selected filter from one category as a link.

Code sample

```
{% include molecule('active-filter-single-select', 'CatalogPage') with {
  data: {
    isMultiSelect: true
  }
} only %}
```

filter-category (molecule)

Displays a list of filter categories, which enables you to filter the categories in a more specific manner. A counter next to each filter item informs you of how many products a filter item contains.

Code sample

```
{% include molecule('filter-category', 'CatalogPage') with {
  data: {
    filter: filter,
    filterPath: filterPath,
    categories: categories,
    isEmptyCategoryFilterValueVisible: isEmptyCategoryFilterValueVisible
  }
} only %}
```

filter-enumeration (molecule)

Displays a list of all filters from one category, in which you can select multiple filters.

Code sample

```
{% include molecule('filter-enumeration', 'CatalogPage') with {
  data: {
    filter: filter,
    isMultiSelect: isMultiSelect
  }
} only %}
```

filter-multi-select (molecule)

Displays a list of all filters from one category, in which you can select multiple filters.

Code sample

```
{% include molecule('filter-multi-select', 'CatalogPage') with {
  data: {
    isMultiSelect: true
  }
} only %}
```

filter-price-range (molecule)

Allows to filter the products by price range with minimum and maximum price manually.

Code sample

```
{% include molecule('filter-price-range', 'CatalogPage') only %}
```

filter-range (molecule)

Allows you to filter manually by range with the minimum and maximum values.

Code sample

```
{% include molecule('filter-range', 'CatalogPage') with {  
  data: {  
    parameter: parameter,  
    min: min,  
    activeMin: activeMin,  
    max: max,  
    activeMax: activeMax  
  }  
} only %}
```

filter-rating (molecule)

Allows you to filter by rating. The values are normalized within boundaries. The maximum value can be set (to max), the minimum is always 0.

Code sample

```
{% include molecule('filter-rating', 'CatalogPage') with {  
  data: {  
    filter: filter  
  }  
} only %}
```

filter-single-select (molecule)

Displays a list of all filters from one category, in which you can select one filter.

Code sample

```
{% include molecule('filter-single-select', 'CatalogPage') with {  
  data: {  
    isMultiSelect: false  
  }  
}
```

```
} only %}
```

sort (molecule)

The sorting section allows you to sort by name, price, or rating, as well as display a product grid in reverse order and an amount of sort results.

Code sample

```
{% include molecule('sort', 'CatalogPage') only %}
```

spelling-suggestion (molecule)

Displays the revised suggested query if the value was entered incorrectly in the search field.

Code sample

```
{% include molecule('spelling-suggestion', 'CatalogPage') with {  
  data: {  
    suggestion: 'suggestion'  
  }  
} only %}
```

suggestion-links-list (molecule)

Displays a list of links (to product or page) in the search drop-down menu if there is a match with the entered value.

Code sample

```
{% include molecule('suggestion-links-list', 'CatalogPage') with {  
  data: {  
    name: 'name',  
    items: items,  
    length: length,  
    isTitleHiddenOnTablet: false,  
    isSuggestion: true  
  }  
} only %}
```

suggestion-product (molecule)

Short display of the item card with image, name, and price in the search drop-down menu.

Code sample

```
{% include molecule('suggestion-product', 'CatalogPage') with {  
  data: {
```

```
        product: product,
        url: url,
        image: null,
        alt: 'alt',
        title: 'title',
        price: 'price',
        originalPrice: 'originalPrice'
    }
} only %}
```

view-mode-switch (molecule)

A layout switch with two viewing modes: grid and list.

Code sample

```
{% include molecule('view-mode-switch', 'CatalogPage') only %}
```

Organisms

active-filter-section (organism)

Displays a list of all selected filters from all categories as links.

Code sample

```
{% include organism('active-filter-section', 'CatalogPage') with {
    data: {
        facets: facets
    }
} only %}
```

filter-section (organism)

Displays a list of all filter categories.

Code sample

```
{% include organism('filter-section', 'CatalogPage') with {
    data: {
        facets: facets,
        filterPath: filterPath,
        categories: categories,
        isEmptyCategoryFilterValueVisible: isEmptyCategoryFilterValueVisible
    }
} only %}
```

CheckoutPage module

Molecules

summary-item (molecule)

Displays a list of products on checkout summary step, which includes name, quantity, options and price.

Code sample

```
{% include molecule('summary-item', 'CheckoutPage') with {
  data: {
    name: item.name,
    quantity: item.quantity,
    price: item.sumPrice | money,
    options: item.productOptions,
    bundleItems: item.bundleItems,
    quantitySalesUnit: item.quantitySalesUnit,
    amountSalesUnit: item.amountSalesUnit,
    amount: item.amount
  }
} only %}
```

summary-overview (molecule)

Displays a subtotal, discounts, shipping, tax, and grand total for list of products on checkout summary step.

Code sample

```
{% include molecule('summary-overview', 'CheckoutPage') with {
  data: data.overview
} only %}
```

summary-product-packaging-unit-default (molecule)

Displays a product summary price without amount on checkout summary step.

Code sample

```
{% include molecule('summary-product-packaging-unit-default', 'CheckoutPage') with {
  data: {
    name: data.name,
    quantity: data.quantity,
    price: data.price,
    quantitySalesUnit: data.quantitySalesUnit
  }
}
```

```
}  
} only %}
```

CmsBlockWidget module

Molecules

product-cms-block (molecule)

Shows CMS-block content.

Code sample

```
{% include molecule('product-cms-block', 'CmsBlockWidget') with {  
  data: {  
    idProductAbstract: data.product.idProductAbstract  
  }  
} only %}
```

CompanyPage module

Molecules

address-delete-message (molecule)

Displays a message that allows you to confirm whether you want to delete an address from the business unit or not.

Code sample

```
{% include molecule('address-delete-message', 'CompanyPage') with {  
  data: {  
    companyUnitAddress: 'companyUnitAddress',  
    companyBusinessUnits: 'companyBusinessUnits',  
    idCompanyBusinessUnit: idCompanyBusinessUnit,  
    cancelUrl: 'cancelUrl'  
  }  
} only %}
```

business-unit-address-item (molecule)

Displays assigned and unassigned addresses on business unit page, which you can edit or delete.

Code sample

```
{% include molecule('business-unit-address-item', 'CompanyPage') with {
  data: {
    address: address,
    idCompanyBusinessUnit: idCompanyBusinessUnit
  }
} only %}
```

business-unit-address-list (molecule)

Displays a list of business unit addresses in the current business unit.

Code sample

```
{% include molecule('business-unit-address-list', 'CompanyPage') with {
  data: {
    field: field,
    addresses: addresses,
    idCompanyBusinessUnit: idCompanyBusinessUnit
  }
} only %}
```

business-unit-chart-item (molecule)

Displays a business unit chart item link.

Code sample

```
{% include molecule('business-unit-chart-item', 'CompanyPage') with {
  data: {
    node: node,
    level: level
  }
} only %}
```

business-unit-delete-message (molecule)

Displays a message that allows you to confirm whether you want to delete the business unit or not.

Code sample

```
{% include molecule('business-unit-delete-message', 'CompanyPage') with {
  data: {
    companyBusinessUnit: companyBusinessUnit
  }
} only %}
```

business-unit-table (molecule)

Displays a business unit details as table with action links (view, edit, delete).

Code sample

```
{% include molecule('business-unit-table', 'CompanyPage') with {
  data: {
    businessUnits: businessUnits,
    actions: {
      view: true,
      update: false,
      delete: false
    }
  }
} only %}
```

company-navigation (molecule)

Displays sub-page company navigation links.

Code sample

```
{% include molecule('company-navigation', 'CompanyPage') with {
  data: {
    activePage: 'activePage'
  }
} only %}
```

display-company-address (molecule)

Displays company address details.

Code sample

```
{% include molecule('display-company-address', 'CompanyPage') with {
  data: {
    address: 'address'
  }
} only %}
```

overview-info (molecule)

Displays company overview information.

Code sample

```
{% include molecule('overview-info', 'CompanyPage') with {
  data: {
    company: 'company',
    defaultBillingAddress: defaultBillingAddress
  }
} only %}
```



```
    }  
  } only %}
```

permission-table (molecule)

Displays a roles permission as table with action links (configure, unassign, assign).

Code sample

```
{% include molecule('permission-table', 'CompanyPage') with {  
  data: {  
    permissions: permissions,  
    idCompanyRole: idCompanyRole,  
    actions: {  
      configure: false,  
      switch: false  
    }  
  }  
} only %}
```

role-table (molecule)

Displays a company roles as table with action links (view, update, delete).

Code sample

```
{% include molecule('role-table', 'CompanyPage') with {  
  data: {  
    roles: data.roles,  
    actions: {  
      view: true,  
      update: false,  
      delete: false  
    }  
  }  
} only %}
```

role-user-table (molecule)

Displays a company user roles as table with action links (unassign, assign).

Code sample

```
{% include molecule('role-user-table', 'CompanyPage') with {  
  data: {  
    users: users,  
    idCompanyRole: idCompanyRole,  
    actions: {  
      switch: true  
    }  
  }  
}
```

```
} only %}
```

user-table (molecule)

Displays a company user details as table with action links (edit, disable, enable, delete).

Code sample

```
{% include molecule('user-table', 'CompanyPage') with {
  data: {
    users: users,
    currentCompanyId: currentCompanyId,
    actions: {
      update: true,
      delete: true
    }
  }
} only %}
```

CompanyUserInvitationPage module

Molecules

invitation-link (molecule)

Displays an innovation link, which redirects you to the company/user invitation page where you can choose an invitation file.

Code sample

```
{% include molecule('invitation-link', 'CompanyUserInvitationPage') only %}
```

invitation-link (molecule)

Displays an invitation links as table, which contains a company users info, and action links (send, resend, delete).

Code sample

```
{% include molecule('invitation-link', 'CompanyUserInvitationPage') with {
  data: {
    invitations: invitations
  }
} only %}
```

CompanyWidget module

Molecules

company-business-unit-address-handler (molecule)

Implements logic to fill form with customer or business address data.

Code sample

```
{% include molecule('company-business-unit-address-handler', 'CompanyWidget') with {
  {
    attributes: {
      'form-selector': '.js-form-handler__' ~ data.formType,
      'trigger-selector': '.js-from-add-new-address__' ~ data.formType,
      'data-selector': '.js-from-select-address__' ~ data.formType,
      'ignore-selector': '.js-form-clear__ignore-element',
      'customer-address-id-selector': '.js-form-customer-id__' ~ data.formType,
      'default-address-selector': '.js-form-default-address__' ~ data.formType,
      'custom-address-trigger': '.js-from-clear__' ~ data.formType ~ '-toggler',
      'addresses': data.addresses
    }
  }
} only %}
```

CustomerPage module

Molecules

customer-navigation (molecule)

Displays sub-page customer navigation links.

Code sample

```
{% include molecule('customer-navigation', 'CustomerPage') with {
  data: {
    activePage: 'activePage',
    activeEntityId: activeEntityId
  }
} only %}
```

order-detail-product-packaging-unit-default (molecule)

Displays a list of order details as table which includes product sku, name, price, quantity, and item total.

Code sample

```
{% include molecule('order-detail-product-packaging-unit-default', 'CustomerPage')
with {
  data: {
    concreteItem: item,
    currencyIsoCode: data.currencyIsoCode,
    concreteItemDefault: concreteItem
  }
} only %}
```

order-detail-table (molecule)

Displays a customer order details as table, which includes products sku, product name, price, quantity, and item total.

Code sample

```
{% include molecule('order-detail-table', 'CustomerPage') with {
  data: {
    items: items,
    currencyIsoCode: 'currencyIsoCode'
  }
} only %}
```

order-summary (molecule)

Displays a summary of customer order as a subtotal, shipment, discount, and grand total.

Code sample

```
{% include molecule('order-summary', 'CustomerPage') with {
  data: {
    order: order
  }
} only %}
```

order-table (molecule)

Displays a customer orders as table, which includes order id, order date, total, and action links (view order, reorder).

Code sample

```
{% include molecule('order-table', 'CustomerPage') with {
  data: {
    orders: orders
  }
} only %}
```

DiscountPromotionWidget module

Molecules

variant (molecule)

Enables, if the product has many variants, displays as a list in which you can choose one of the options.

Code sample

```
{% include molecule('variant', 'DiscountPromotionWidget') with {
  data: {
    name: name,
    formName: 'attributes[' ~ data.idProductAbstract ~ '][' ~ name ~ ']',
    values: values,
    selectedValue: selectedValue,
    label: ('product.attribute.' ~ name) | trans,
    isAvailable: isAvailable
  }
} only %}
```

variant-configurator (molecule)

Displays a list of product variants as select element and if variants is already selected - as static list with reset button.

Code sample

```
{% include molecule('variant-configurator', 'DiscountPromotionWidget') with {
  data: {
    idProductAbstract: data.idProductAbstract,
    superAttributes: data.superAttributes,
    selectedAttributes: data.selectedAttributes,
    availableAttributes: data.availableAttributes
  }
} only %}
```

DiscountWidget module

Molecules

cart-discount-summary (molecule)

Displays a list of vouchers with remove link and discount rules.

Code sample

```
{% include molecule('cart-discount-summary', 'DiscountWidget') with {
  data: {
    voucherDiscounts: data.cart.voucherDiscounts,
    ruleDiscounts: data.cart.cartRuleDiscounts,
    discountTotal: data.cart.totals.discounttotal
  }
} only %}
```

MultiCartPage module

Molecules

quote-table (molecule)

Displays table with list of shopping carts with short description.

Code sample

```
{% include molecule('quote-table', 'MultiCartPage') with {
  data: {
    quotes: data.quotes,
    actions: {
      update: true,
      delete: true,
      duplicate: true
    },
    isQuoteDeletable: data.isQuoteDeletable
  }
} only %}
```

MultiCartWidget module

Molecules

add-to-cart-selector (molecule)

Displays a drop-down with list of shopping carts.

Code sample

```
{% include molecule('add-to-cart-selector', 'MultiCartWidget') with {
  data: {
    carts: carts,
    disabled: false
  }
}
```

```
} only %}
```

cart-table (molecule)

Displays a shopping cart as table with action links (view, update, delete).

Code sample

```
{% include molecule('cart-table', 'MultiCartWidget') with {
  data: {
    carts: carts,
    widgets: widgets,
    actions: {
      view: true,
      update: false,
      set_default: true,
      delete: true
    }
  }
} only %}
```

cart-table-detail (molecule)

Displays a shopping cart details as a table row.

Code sample

```
{% include molecule('cart-table-detail', 'MultiCartWidget') with {
  data: {
    cart: cart,
    actions: actions
  }
} only %}
```

mini-cart-detail (molecule)

Displays a drop-down at the top navigation with list of shopping carts, which includes cart name, number of items in cart, price mode, and cart price.

Code sample

```
{% include molecule('mini-cart-detail', 'MultiCartWidget') with {
  data: {
    cart: cart
  }
} only %}
```

mini-cart-radio (molecule)

Displays as a switcher for list of shopping carts, which sets an active cart for checkout.

Code sample

```
{% include molecule('mini-cart-radio', 'MultiCartWidget') with {
  data: {
    idCart: idCart,
    isDefault: isDefault,
  }
} only %}
```

PriceProductVolumeWidget module

Molecules

volume-price (molecule)

Displays a product price.

Code sample

```
{% include molecule('volume-price', 'PriceProductVolumeWidget') with {
  data: {
    amount: data.amount | money,
    originalAmount: data.originalAmount
  }
} only %}
```

volume-price-table (molecule)

Displays a product price range table.

Code sample

```
{% include molecule('volume-price-table', 'PriceProductVolumeWidget') with {
  data: {
    volumePrices: data.volumePrices
  }
} only %}
```

ProductAlternativeWidget module

Molecules

alternative-products-table (molecule)

Displays a table with alternative products limited by maxShownItems option.

Code sample

```
{% include molecule('alternative-products-table', 'ProductAlternativeWidget') with {
  data: {
    item: data.item,
    shoppingList: data.shoppingList,
    products: data.products
  }
} only %}
```

ProductDetailPage module

Molecules

product-carousel (molecule)

Creates a carousel with product images using the simple-carousel component; also shows the product labels.

Code sample

```
{% include molecule('product-carousel', 'ProductDetailPage') with {
  data: {
    product: data.product
  }
} only %}
```

product-configurator (molecule)

Shows a block with the most important product information and provides ability to configure product using the variant-configurator component. Also has the quantity drop-down and the add to cart button.

Code sample

```
{% include molecule('product-configurator', 'ProductDetailPage') with {
  data: {
    product: data.product
  }
} only %}
```

product-detail (molecule)

Shows a list of product attributes.

Code sample

```
{% include molecule('product-detail', 'ProductDetailPage') with {
  class: 'box',
  data: {
    description: data.product.description,
    attributes: data.product.attributes
  }
} only %}
```

variant (molecule)

Shows a drop-down with product variants as options in it. Reloads a page when the option is changed.

Code sample

```
{% include molecule('variant', 'ProductDetailPage') with {
  data: {
    name: 'name',
    formName: 'formName',
    values: values,
    selectedValue: 'selectedValue',
    label: 'label',
    isAvailable: true
  }
} only %}
```

variant-configurator (molecule)

Shows a list of the variant drop-downs optionally wrapped with the form tag.

Code sample

```
{% include molecule('variant-configurator', 'ProductDetailPage') with {
  data: {
    superAttributes: data.product.attributeMap.superAttributes,
    selectedAttributes: data.product.selectedAttributes,
    availableAttributes: data.product.availableAttributes,
    useExternalForm: true
  }
} only %}
```

ProductLabelWidget module

Molecules

flag (molecule)

Displays label with short text and icon.

Code sample

```
{% include molecule('flag', 'ProductLabelWidget') with {  
  data: {  
    text: ''  
  }  
} only %}
```

label-group (molecule)

Displays list of labels (with text and icon or only with text).

Code sample

```
{% include molecule('label-group', 'ProductLabelWidget') with {  
  data: {  
    labels: []  
  }  
} only %}
```

tag (molecule)

Displays label with short text.

Code sample

```
{% include molecule('tag', 'ProductLabelWidget') with {  
  data: {  
    text: ''  
  }  
} only %}
```

ProductOptionWidget module

Molecules

product-options (molecule)

Shows the list of product options in "value: price" format.

Code sample

```
{% include molecule('product-options', 'ProductOptionWidget') with {
  data: {
    options: data.cartItem.productOptions
  }
} only %}
```

shopping-list-product-option-list (molecule)

Displays a list of product options as drop-downs on shopping list edit page.

Code sample

```
{% include molecule('shopping-list-product-option-list', 'ProductOptionWidget')
with {
  data: {
    options: data.options
  }
} only %}
```

ProductReviewWidget module

Molecules

new-review-form (molecule)

Displays review form with summary, rating and description fields.

Code sample

```
{% include molecule('new-review-form', 'ProductReviewWidget') with {
  data: {
    form: data.form
  }
} only %}
```

new-review-form (molecule)

Displays rating indicated by stars (usually 1-5), the highest number of stars indicating the best quality etc.

Code sample

```
{% include molecule('rating-selector', 'ProductReviewWidget') with {
  data: {
    value: data.value,
    maxValue: data.maxValue,
    useHalfSteps: true
  },
  attributes: {
    readonly: true
  }
} only %}
```

review (molecule)

Displays text of review, which include title, rating, date, and description.

Code sample

```
{% include molecule('review', 'ProductReviewWidget') with {
  data: {
    summary: review.summary,
    ratingValue: review.ratingValue,
    ratingMaxValue: data.ratingMaxValue,
    nickname: review.nickname,
    createdAt: review.createdAt | formatDate,
    description: review.description
  }
} only %}
```

review-average-display (molecule)

Displays an average rating based on 5-option voting. The average rating calculator takes a number of votes for each option (1 - 5 stars) and gives you the mean rating.

Code sample

```
{% include molecule('review-average-display', 'ProductReviewWidget') with {
  data: {
    summary: data.summary,
    ratingMaxValue: data.ratingMaxValue
  }
} only %}
```

review-distribution-display (molecule)

Displays as a percentage how much each assessment takes.

Code sample

```
{% include molecule('review-distribution-display', 'ProductReviewWidget') with {
  data: {
    summary: data.summary
  }
} only %}
```

Organisms

review-summary (organism)

Displays a section with a list of reviews, average, and detailed rating.

Code sample

```
{% include organism('review-summary', 'ProductReviewWidget') with {
  data: {
    reviews: data.reviews,
    summary: data.summary,
    ratingMaxValue: data.ratingMaxValue,
    hasCustomer: data.hasCustomer,
    pagination: data.pagination
  }
} only %}
```

QuickOrderPage module

Molecules

quick-order-form (molecule)

Displays a form with several rows in which includes SKU and quantity fields. Orders with multiple SKUs can be entered manually, also user can add/remove rows.

Code sample

```
{% include molecule('quick-order-form', 'QuickOrderPage') with {
  data: {
    form: data.forms.items
  }
} only %}
```

quick-order-form-field-list (molecule)

Displays a list of form rows in which includes SKU and quantity fields.

Code sample

```
{% include molecule('quick-order-form-field-list', 'QuickOrderPage') with {
  data: {
    items: data.form.items
  }
} only %}
```

SalesOrderThresholdWidget module

Molecules

sales-order-threshold-expense (molecule)

Displays a threshold expense name and price if order sum less than acceptable and the soft threshold is present.

Code sample

```
{% include molecule('sales-order-threshold-expense', 'SalesOrderThresholdWidget')
with {
  data: {'expense': expense}
} only %}
```

SharedCartPage module

Molecules

user-share-list (molecule)

Displays a share list of user, which you can share shopping cart with certain rights.

Code sample

```
{% include molecule('user-share-list', 'SharedCartPage') with {
  data: {
    shareDetailsForm: shareDetailsForm,
    shared: shared
  }
} only %}
```

SharedCartWidget module

Molecules

shared-cart-add-product-as-separate-item (molecule)

Displays a simple checkbox with text on PDP. When you add a product to shopping cart, if this checkbox is checked, the product will add as separate item.

Code sample

```
{% include molecule('shared-cart-add-product-as-separate-item', 'SharedCartWidget')
ignore missing only %}
```

shared-cart-customers-table (molecule)

Displays a shared carts as table, which you can unshare from another customers or business units.

Code sample

```
{% include molecule('shared-cart-customers-table', 'SharedCartWidget') with {
  data: {
    cart: cart
  }
} only %}
```

ShoppingListNoteWidget module

Molecules

shopping-list-note-toggler (molecule)

Displays a simple link which on click toggle text area field.

Code sample

```
{% include molecule('shopping-list-note-toggler', 'ShoppingListNoteWidget') with {
  data: {
    shoppingListItemNoteForm: form
  }
} only %}
```


ShoppingListPage module

Molecules

share-list (molecule)

Displays a list of business units that you can share your shopping list.

Code sample

```
{% include molecule('share-list', 'ShoppingListPage') with {
  data: {
    shareUnits: data.shareUnits,
    shareForm: data.shareForm
  }
} only %}
```

share-list-item (molecule)

Displays a list item of business unit which shows company name and drop-down with access variants.

Code sample

```
{% include molecule('share-list-item', 'ShoppingListPage') with {
  data: {
    name: user,
    shareForm: data.shareForm[key]
  }
} only %}
```

shopping-list-action-navigation (molecule)

Displays a list of navigation actions which includes edit, share, print, and delete links.

Code sample

```
{% include molecule('shopping-list-action-navigation', 'ShoppingListPage') with {
  data: {
    hasWritePermission: hasWritePermission,
    idShoppingList: shoppingList.idShoppingList,
    shoppingList: shoppingList
  }
} only %}
```

shopping-list-info (molecule)

Displays a short info about owner, access, and number of users or business units whom access was

shared.

Code sample

```
{% include molecule('shopping-list-info', 'ShoppingListPage') with {
  data: {
    shoppingList: data.shoppingList
  }
} only %}
```

shopping-list-message (molecule)

Displays a warning message if you are trying to delete shopping list.

Code sample

```
{% include molecule('shopping-list-message', 'ShoppingListPage') with {
  data: {
    title: data.title,
    backUrl: data.backUrl,
    idShoppingList: data.shoppingList.idShoppingList,
    sharedCompanyUsers: data.sharedCompanyUsers,
    sharedCompanyBusinessUnits: data.sharedCompanyBusinessUnits
  }
} only %}
```

shopping-list-navigation (molecule)

Displays a navigation list of shopping lists as links.

Code sample

```
{% include molecule('shopping-list-navigation', 'ShoppingListPage') with {
  data: {
    activeShoppingListId: data.activeShoppingListId,
    shoppingListCollection: data.shoppingListCollection
  }
} only %}
```

shopping-list-overview-table (molecule)

Displays a list of shopping lists as table which includes name, owner, date of creation, access, number of items, and actions links.

Code sample

```
{% include molecule('shopping-list-overview-table', 'ShoppingListPage') with {
  data: {
    shoppingLists: data.shoppingListCollection.shoppingLists,
```

```

        enableTableForm: true,
        shoppingListResponse: data.shoppingListResponse
    }
} only %}

```

shopping-list-overview-update-table (molecule)

Displays a list of shopping lists as table which includes name, owner, date of creation, access, number of items, and actions links.

Code sample

```

{% include molecule('shopping-list-overview-update-table', 'ShoppingListPage') with {
  data: {
    shoppingListItems: data.shoppingList.items,
    idShoppingList: data.idShoppingList,
    shoppingList: data.shoppingList,
    shoppingListItemProducts: data.shoppingListItemProducts,
    form: data.form
  }
} only %}

```

shopping-list-permission (molecule)

Displays a permission for shopping lists as 'Full access' or 'Read only'.

Code sample

```

{% include molecule('shopping-list-permission', 'ShoppingListPage') with {
  data: {
    hasWritePermission: hasWritePermission
  }
} only %}

```

shopping-list-table (molecule)

Displays a list of product in shopping list as table which includes product name, price, quantity, availability, and action link.

Code sample

```

{% include molecule('shopping-list-table', 'ShoppingListPage') with {
  data: {
    shoppingList: data.shoppingList,
    shoppingListItems: data.shoppingListItems
  }
} only %}

```

ShoppingListWidget module

Molecules

cart-to-shopping-list (molecule)

Displays a link which on click add shopping cart to shopping list.

Code sample

```
{% include molecule('cart-to-shopping-list', 'ShoppingListWidget') ignore missing
with {
    data: {
        quote: quote
    }
} only %}
```

form-data-injector (molecule)

Copies a form data to hidden form feilds on submit.

Code sample

```
{% include molecule('form-data-injector', 'ShoppingListWidget') with {
    attributes: {
        'destination-form-selector': '.js-shopping-list__form',
        'fields-selector': '.js-product-configurator__form'
    }
} only %}
```

shop-list-item (molecule)

Displays a drop-down at top navigation menu with list of shopping lists which includes shopping list name, customer name, number of products in list, and access.

Code sample

```
{% include molecule('shop-list-item', 'ShoppingListWidget') with {
    data: {
        shoppingList: shoppingList
    }
} only %}
```

WishlistPage module

Molecules

add-wishlist (molecule)

Displays a form to add a new wishlist which includes text field, and submit button.

Code sample

```
{% include molecule('shop-list-item', 'ShoppingListWidget') with {
  data: {
    shoppingList: shoppingList
  }
} only %}
```

wishlist-detail-table (molecule)

Displays a list of product in wishlist as table which includes product details, price, availability, and action links.

Code sample

```
{% include molecule('wishlist-detail-table', 'WishlistPage') with {
  data: {
    wishlistItems: data.wishlistItems,
    wishlistName: data.wishlistName
  }
} only %}
```

wishlist-overview-table (molecule)

Displays a list of wishlists as table which includes wishlist name, number of items in wishlist, date of creation, and action links.

Code sample

```
{% include molecule('wishlist-overview-table', 'WishlistPage') with {
  data: {
    wishlists: data.wishlists,
    wishlistForm: data.wishlistForm
  }
} only %}
```

WishlistWidget module

Molecules

wishlist-selector (molecule)

Displays a form with two fields (drop-down and button), which on click on the button will add the product to the wishlist, selected from the drop-down.

Code sample

```
{% include molecule('wishlist-selector', 'WishlistWidget') with {
  data: {
    sku: data.sku,
    idProductConcrete: data.idProductConcrete
  }
} only %}
```

wishlist-selector-for-pdp (molecule)

Displays a form with two fields (drop-down and button), which on click on the button will add the product to the wishlist, selected from the drop-down.

Code sample

```
{% include molecule('wishlist-selector-for-pdp', 'WishlistWidget') ignore missing
with {
  data: {
    sku: data.product.sku,
    idProductConcrete: data.product.idProductConcrete | default
  }
} only %}
```

ProductSetWidget module

Organisms

product-set-cms-content (organism)

Displays a CMS section, which includes text messages, simple carousel with images, list of products, which you can add to your shopping cart.

Code sample

```
{% include organism('product-set-cms-content', 'ProductSetWidget') ignore missing
with {
```

```
data: {
  products: productViews,
  name: productSet.name,
  description: productSet.description,
  images: productSet.imageSets | default([])
}
} only %}
```

product-set-details (organism)

Displays a CMS section, which includes text messages, simple carousel with images, list of products, which you can add to your shopping cart.

Code sample

```
{% include organism('product-set-details', 'ProductSetWidget') ignore missing with
{
  data: {
    products: data.views,
    name: data.set.name,
    description: data.set.description,
    images: data.set.imageSets | default([])
  }
} only %}
```