



# Storeden

Template Development

<https://themes-developers.storeden.com>

v.2019.1

## Cosa vedremo

1. Com'è fatto un tema Storeden
2. Schema.json
3. Preset.json
4. Stili custom sui widget
5. Lang\_ISO.json
6. Storeden.js
7. Funzioni Twig+Storeden

1.

# tema Storeden

Componenti base

# Live preview

[Home](#) [Negozio](#) [Donna](#) [Uomo](#) [Bambina](#) [Bambino](#) [Scarpe](#) [Chi siamo](#) [Contattaci](#) [Blog](#)



Site preview

Backoffice

Sections

General



Header

Image slider

H1 seo content

Featured products

Banner image

Tagged products

Add new section

Footer



## Fashion Store, abbigliamento per tutti i gusti

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Personalizzare un tema

Configurazioni  
.json

Html - CSS-  
Javascript

Interfaccia  
live config



# Struttura di un tema

- ◆ Twig + Storenden
  - Guida: <https://twig.sensiolabs.org/doc/2.x/>
  - [Lista delle funzioni disponibili](#)
- ◆ Bootstrap 3
  - Guida: <http://getbootstrap.com/>
- ◆ Storenden.js
  - Guida [storenden.js](#)
- ◆ HTML 5
- ◆ JQuery 2.2.4
- ◆ Font Awesome
  - Icons: <http://fontawesome.io/icons/>

# Directory di un tema

## ◆ Ecommerce > Tema > Modifica html/css

/theme

/config

/template

/assets

/locales

### Essential Theme

Utilizza i link di seguito per configurare e modificare il template del tuo sito.

✓ Tema in uso

📘 Guida alla configurazione dello storefront

<> Modifica html/css

🖌 Personalizza

👁 Visualizza

🗣 Traduzioni

# THEME - ASSETS - LOCALES

## THEME / theme.style

File CSS del tema, ha accesso alla variabile **theme** che contiene le configurazioni

### THEME

theme.style

## ASSETS / script.js

script necessari al funzionamento del tema

### ASSET

script.js

## LOCALES / LANG\_ISO.json

File di traduzione, per le chiavi di sistema o chiavi personalizzate. Si usa per le “etichette” del sito.

### LOCALES

de\_DE.json  
en\_US.json  
es\_ES.json  
fr\_FR.json  
it\_IT.json



# TEMPLATE / skeleton.html

## \*.html

I file HTML del tema sono tutti contenuti nella directory Template.

Il file "index" è skeleton.html che include il template corretto in base al path corrente.

```
{% include request.page %}
```

[www.iltuosito.com/product/1330477/t-shirt-rosso](http://www.iltuosito.com/product/1330477/t-shirt-rosso)

```
{{ request | var_dump }}
```

```
{
  "page": "product.html",
  "url": "product",
  "script_name": "",
  "path": [
    "",
    "product",
    "1330477",
    "t-shirt-rosso"
  ],
  "query": [ ... ],
  "payload": [ ... ],
  "method": "GET"
}
```

# TEMPLATE

## Tipologie di template

- ◆ Di sistema (shop.html, product.html, cart.html, ... )
- ◆ Custom del tema (header.html, footer.html, product\_preview.html, ... )
- ◆ Preset `template_PRESET_NAME.html`

# TEMPLATE + TWIG

## Alcuni esempi di istruzioni twig

```
{% include "filename.html" %}
```

```
{{ variable }}
```

```
{% set variable = value %}
```

```
{% set variable = [] %}
```

```
{% if test %} ... {% else %} ... {% endif %}
```

```
{% for element in elements %} ... {% endfor %}
```

```
{{ variable | var_dump }}
```

- ◆ <https://twig.sensiolabs.org/doc/2.x/>
- ◆ <https://themes-developers.storenden.com/hc/it/articles/360019784611-Indice-funzioni-twig->

# Directory CONFIG

## config.json

Personalizzazioni del tema.  
Sono accessibili nei file del  
tema tramite la variabile  
**theme**

## storeden.json

Informazioni del tema: id,  
titolo, versione live  
configuration

## preset.json

Insieme dei componenti  
personalizzabili del tema

## schema.json

Configurazioni base e di  
pagina del tema

### **CONFIG**

config.json

preset.json

schema.json

storeden.json

# Variabili utili

## **theme**

Ultime personalizzazioni del tema memorizzate: widget, configurazioni, ...

## **request**

Tutti i dati relativi alla request corrente

## **locale**

Codice ISO della lingua corrente

## **user**

Dati dell'utente che ha effettuato il login, null altrimenti

**Utilizzabili in tutte le pagine del tema**

# Altre variabili utili

## shop.html

- ◆ **product\_count**      numero totale prodotti (non paginati)
- ◆ **productlist**      prodotti della pagina corrente dello shop

## cart.html

- ◆ **cart\_id**              id del carrello corrente

2.

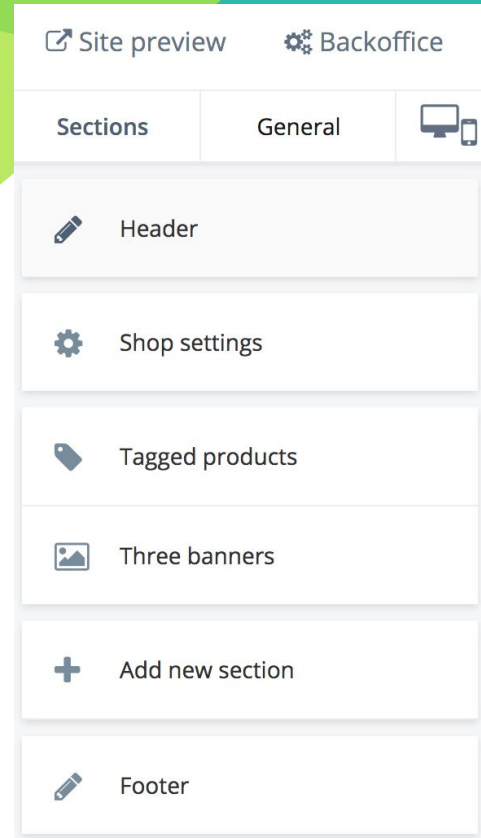
# schema.json

Configurazioni generali, header, footer, pagine

# CONFIG - schema.json

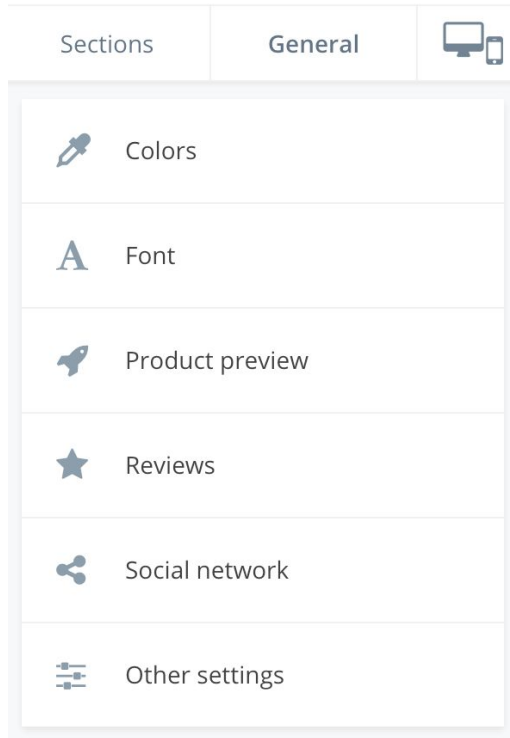
- ◆ Configurazioni generali del tema: font, colori, preview prodotto, ...
- ◆ Header
- ◆ Footer
- ◆ Configurazioni di pagina \*
  - ◆ homepage
  - ◆ shop
  - ◆ product
  - ◆ cart
  - ◆ page
  - ◆ about-us
  - ◆ contact-us
  - ◆ blog
  - ◆ post

\* Le configurazioni hanno effetto su tutte le pagine dello stesso tipo.



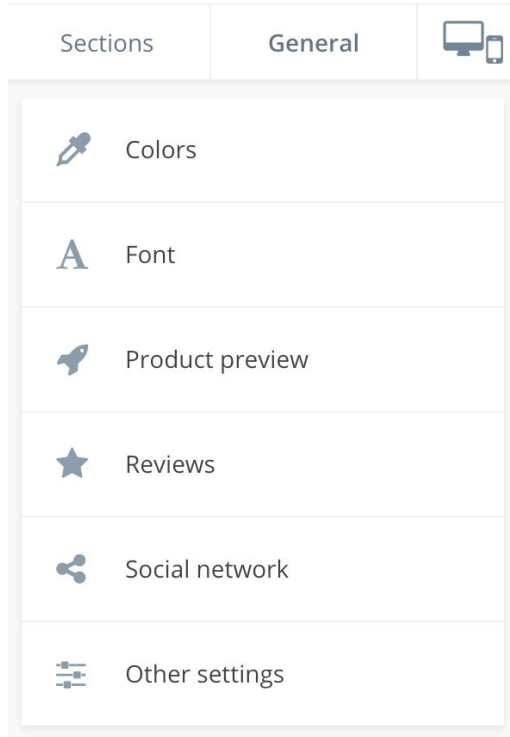


# CONFIG - schema.json



```
{  
  "general" : [ ... ],  
  "header": { ... },  
  "footer" : { ... },  
  "pages":{  
    "shop": { ... },  
    "product": { ... },  
    "cart": { ... },  
    "blog": { ... },  
    "post": { ... },  
    ...  
  }  
}
```

# CONFIG - schema.json - general



```
{  
  "general" : [  
    {  
      "Label" : "Colors",  
      "id" : "colors",  
      "icon" : "fa-eyedropper",  
      "settings" : [ ... ]  
    }, ...  
  ], ...  
}
```

**theme.settings.general.colors**

# CONFIG - schema.json - general - colors

< Colors

**BASIC** ^

Background	#FFFFFF	<input type="color" value="#FFFFFF"/>
Text color	#555555	<input type="color" value="#555555"/>
Link color	#698EB8	<input type="color" value="#698EB8"/>

**BUTTONS** ^

**BORDERS** ^

```
"settings" : [  
  {  
    "label" : "Basic",  
    "fields" : [  
      {  
        "type" : "color",  
        "id" : "body_background",  
        "label" : "Background",  
        "default" : "#FFF"  
      },  
      ...  
    ]  
  }  
]
```

# CONFIG - schema.json - general - colors

Per accedere al valore una volta salvata la configurazione tramite live preview

```
theme.settings.general.colors.body_background
```

I campi di tipo color sono “light” cioè le modifiche avvengono live senza il reload della pagina di preview. Nel CSS quindi utilizzando twig e la variabile **theme**:

```
.body { background-color: {{ theme_value(theme.settings.general.colors.body_background, "settings.general.colors.body_background") }}; }
```

# CONFIG - schema.json - header

< Header

HEADER ^		
Background	#FFFFFF	<input type="color"/>
Text color	#333333	<input type="color"/>
Link color	#698EB8	<input type="color"/>
STORE LOGO v		
CART BADGE v		
TOP MESSAGE v		
MENU v		

```
"header": {  
  "label": "Header",  
  "id": "header",  
  "icon": "fa-pencil",  
  "settings": [ ... ]  
}  
  
"footer" : {  
  "label": "Footer",  
  "id": "footer",  
  "icon": "fa-pencil",  
  "settings": [ ... ]  
}
```

# CONFIG - schema.json - header

```
theme.settings.header
```

Spedizioni gratis in tutta Italia! Approfittane subito

**FASHION STORE**



[Home](#) [Negozio](#) [Donna](#) ▾ [Uomo](#) ▾ [Bambina](#) ▾ [Bambino](#) [Scarpe](#) [Chi siamo](#) [Contattaci](#) [Blog](#)


# CONFIG - schema.json - header - top message - yes\_no


```
"settings": [ {  
  "label" : "Top message",  
  "fields" : [  
    {  
      "type" : "yes_no",  
      "id" : "show_top_message",  
      "label" : "Enable top message",  
      "multilang" : "yes",  
      "default" : "yes"  
    }  
  ]  
}, ... ]
```

< Header

CART BADGE ▾


TOP MESSAGE ▲

 Enable top message


 Text

Background color #698EB8

Text color #FFFFFF

 Link

Remove

 Title (on mouse over) ⓘ

Link relation ⓘ

Link behaviour ⓘ

# CONFIG - schema.json - header - top message - yes\_no

## Se non multilingua:

```
theme.settings.header.show_top_message
```

## Se multilingua:

```
attribute(theme.settings.header,"show_top_message_"~locale)
```

```
{% if theme.settings.header.show_top_message == "yes" %}
```

Oppure

```
{% if attribute(theme.settings.header,"show_top_message_"~locale) == "yes" %}
```

```
<div id="top_message">
```

```
    {# here goes my top message text #}
```

```
</div>
```

```
{% endif %}
```

## < Header

CART BADGE

TOP MESSAGE

 Enable top message

 Text

tis in tutta Italia! Approfittane subito

Background color #698EB8

Text color #FFFFFF

 Link

/shop

Remove

 Title (on mouse over) ⓘ

Link relation ⓘ

Link behaviour ⓘ

Open in the same page



# CONFIG - schema.json - header


## - top message - text


```
"settings": [ {  
  "label" : "Top message",  
  "fields" : [  
    {  
      "type" : "text",  
      "id" : "top_message_text",  
      "label" : "Text",  
      "multilang" : "yes",  
      "default" : {  
        "it_IT" : "Spedizioni gratis!",  
        "en_US" : "Free shipping!"  
      }  
    }  
  ]  
}, ... ]
```

< Header

CART BADGE ▾


TOP MESSAGE ▲

 Enable top message


 Text

Background color #698EB8

Text color #FFFFFF

 Link

Remove

 Title (on mouse over) ⓘ

Link relation ⓘ

Link behaviour ⓘ

# CONFIG - schema.json - header - top message - text

## Se non multilingua:

```
theme.settings.header.top_message_text
```

## Se multilingua:


```
attribute(theme.settings.header,"top_message_text_"~locale)
```

```
{% if attribute(theme.settings.header,"show_top_message_"~locale) == "yes" %}  
  <div id="top_message"  
    data-editor="settings_header_top_message_text_{{locale}}" >  
    {{ attribute(theme.settings.header,"top_message_text_"~locale) }}  
  </div>  
{% endif %}
```

## < Header

CART BADGE ▾

TOP MESSAGE ▴

 Enable top message

 Text

tis in tutta Italia! Approfittane subito

Background color #698EB8

Text color #FFFFFF

 Link

/shop

Remove

 Title (on mouse over) ⓘ

Link relation ⓘ

---

Link behaviour ⓘ

Open in the same page

# CONFIG - schema.json - header


## - top message - link


```
"settings": [ {  
  "label": "Top message",  
  "fields": [  
    {  
      "type": "link",  
      "id": "top_message_link",  
      "label": "Link",  
      "multilang": "yes",  
      "default": {  
        "it_IT": "/shop",  
        "en_US": "/shop"  
      }  
    }  
  ]  
}, ... ]
```

< Header

CART BADGE ▾


TOP MESSAGE ▲

 Enable top message


 Text

Background color #698EB8

Text color #FFFFFF

 Link

Remove

 Title (on mouse over) ⓘ

Link relation ⓘ

Link behaviour ⓘ

# CONFIG - schema.json - header - top message - link

CAMPI automatici: TITLE (multilang), REL, TARGET

## Se non multilingua:

```
theme.settings.header.top_message_link  
theme.settings.header.top_message_link_TITLE  
theme.settings.header.top_message_link_REL  
theme.settings.header.top_message_link_TARGET
```

## Se multilingua:

```
attribute(theme.settings.header,"top_message_link_"~locale)  
attribute(theme.settings.header,"top_message_link_TITLE_"~locale)  
theme.settings.header.top_message_link_REL  
theme.settings.header.top_message_link_TARGET
```

 Link

Remove

 Title (on mouse over) 

Link relation 

---

Link behaviour 

Open in the same page

# CONFIG - schema.json - header - top message

```
{% if attribute(theme.settings.header,"show_top_message_"~locale) == "yes" %}
  <div id="top_message">
    <a href="{{attribute(theme.settings.header, 'top_message_link_'~locale)}}"
      rel="{{theme.settings.header.top_message_link_REL}}"
      target="{{theme.settings.header.top_message_link_TARGET}}"
      title="{{attribute(theme.settings.header, 'top_message_link_TITLE_'~locale)}}" >

      <span data-editor="settns_header_top_message_text_{{locale}}" >
        {{ attribute(theme.settings.header,"top_message_text_"~locale) }}
      </span>
    </a>
  </div>
{% endif %}
```

# CONFIG - schema.json - footer

theme.settings.footer

## Fashion Store

Fashion Store SRL  
Piazza Duomo, Treviso 31100 TV  
P.IVA 123456789123

## Informazioni

[Resi e cambio merce](#)  
[I metodi di pagamento](#)  
[Spedizioni](#)  
[Hai bisogno di aiuto?](#)

## Personalizza titolo

[Home](#)  
[L'azienda](#)  
[News](#)  
[Contatti](#)

Copyright © 2017 Fashion Store. All rights reserved. [Informativa Privacy](#) [Condizioni di Vendita](#) [Informativa Cookies](#)

Powered by [Storeden](#)

# CONFIG - schema.json - footer

## < Footer

### BASIC

Background #FFFFFF

Text color #333333

Link color #698EB8

### STORE INFORMATION

### FIRST LINKS GROUP

### SECOND LINKS GROUP

```
"footer" : {  
  "label": "Footer",  
  "id": "footer",  
  "icon": "fa-pencil",  
  "settings": [ ... ]  
}
```

### Personalizza titolo

[Home](#)  
[Lazienda](#)  
[News](#)  
[Contatti](#)


# CONFIG - schema.json - footer - navigation

## ◆ Ecommerce > Gestione Menu > Aggiungi menu


Elenco dei link

Inserisci qui sotto i link per creare i tuoi menù di navigazione.

+ Aggiungi link alla navigazione

Etichetta	Tipologia link	
<input type="text" value="Home"/>	<input type="text" value="URL"/>	<input type="text" value="/homepage"/>
Relazione link	Comportamento link	 <b>Elimina</b>
<input type="text" value="--"/>	<input type="text" value="--"/>	

---

Etichetta	Tipologia link	
<input type="text" value="L'azienda"/>	<input type="text" value="URL"/>	<input type="text" value="/about-us"/>
Relazione link	Comportamento link	 <b>Elimina</b>
<input type="text" value="--"/>	<input type="text" value="--"/>	

→

**Personalizza titolo**

- Home
- L'azienda
- News
- Contatti



# CONFIG - schema.json - footer - navigation

```
"settings": [ {  
  "label" : "Second links group",  
  "fields" : [  
    {  
      "type" : "navigation",  
      "id" : "box_2_menu",  
      "label" : "Select custom menu",  
      "multilang" : "yes",  
      "default" : {}  
    }  
  ]  
}, ... ]
```

## Personalizza titolo

[Home](#)  
[L'azienda](#)  
[News](#)  
[Contatti](#)

## < Footer


BASIC ▾

STORE INFORMATION ▾


FIRST LINKS GROUP ▾

SECOND LINKS GROUP ▴

Hide this content

 Section title

Personalizza titolo

 Select custom menu

Top menu - fashion week theme ▾

# CONFIG - schema.json - footer - navigation

## Se non multilingua:

```
{% set menuName = theme.settings.footer.box_2_menu %}
```

## Se multilingua:

```
{% set menuName = attribute(theme.settings.footer, 'box_2_menu_' ~ locale) %}
```

```
{% set menulist = getNavigation(menuName) %}
```

```
{% include 'menulist.html' %}
```

Il template menulist.html genera i link del menu selezionato utilizzando la variabile menulist.

# CONFIG - schema.json - pages

- ◆ Si attivano quando si naviga nella live preview in una pagina configurabile
- ◆ Le pagine configurabili sono quelle che rispondono a url definiti dal sistema
- ◆ L'id deve corrispondere al primo frammento del path

Esempio:

/shop/search/t-shirt → shop

/product/234/t-shirt → product


...







```
"page" : [  
  "homepage": {},  
  "shop": {},  
  "product": {},  
  "cart": {},  
  "blog": {},  
  "post": {}  
]
```

**theme.settings.PAGE\_ID**

# CONFIG - schema.json - pages - shop

[Site preview](#) [Backoffice](#)

Sections    General    

-  Header
-  Shop settings
-  Tagged products
-  Three banners
-  Add new section
-  Footer

```
"shop" : {  
  "label" : "Shop settings",  
  "id" : "shop",  
  "icon" : "fa-gear",  
  "settings" : {  
    ...  
  }  
}
```

# CONFIG - schema.json - pages - shop - select

## < Shop settings

CATEGORY AND BRAND SETTINGS ▾

Show categories navigati...

Show brands navigation

Category image width ⓘ

Aligned with text ▾

```
"shop" : {
  "label" : "Shop settings",
  "id" : "shop",
  "icon" : "fa-gear",
  "settings" : {
    {
      "label" : "Category & Brand",
      "fields" : [ ... ]
    },
    ...
  }
}
```

# CONFIG - schema.json - pages - shop - select

```
"fields" : [  
  {  
    "type" : "select",  
    "id" : "shop_image_width",  
    "label" : "Category/brand image with",  
    "info" : "Some info here... ",  
    "default" : "full",  
    "options" : {  
      {  
        "option" : "full",  
        "label" : "Full width"  
      },  
      {  
        "option" : "aligned",  
        "label" : "Aligned with text"  
      }  
    }  
  }  
]
```

## < Shop settings

### CATEGORY AND BRAND SETTINGS ▾

Show categories navigati...

Show brands navigation

Category image width ⓘ

Aligned with text ▾

`theme.settings.shop.shop_image_width`

3.

# preset.json

Definizione di widget



## WIDGET o PRESET

Elemento che compone un tema, può essere aggiunto spostato o eliminato ed è composto da un template html e un insieme di campi di configurazioni definiti nel file preset.json



# Esempio

Paragrafo di testo con immagine, la quale può essere allineata a destra o sinistra oppure disabilitata.



## Scopri chi siamo...

Lorem ipsum dolor sit amet, **consectetur adipiscing elit**, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# Esempio

## Elenco di prodotti in primo piano

### Prodotti in vetrina



Abito in seta bianco stile greco con decorazioni oro

€ 290,00

[Aggiungi al carrello](#)

[♡ Wishlist](#)

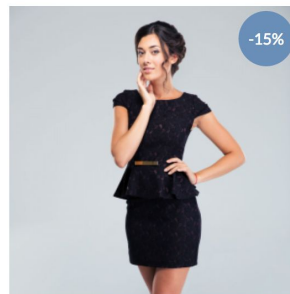


Abito corto in cotone blu con pois bianchi

€ 90,00

[Aggiungi al carrello](#)

[♡ Wishlist](#)

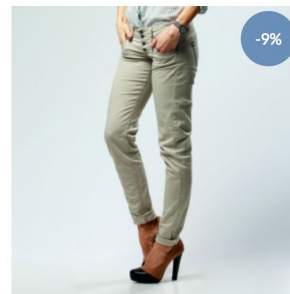


Abito da cerimonia blu fantasia jaquard con cinturino in vita

€ 300,00 € 255,00

[Aggiungi al carrello](#)

[♡ Wishlist](#)



Pantaolone in cotone lungo colore beige

€ 110,00 € 100,00

[Aggiungi al carrello](#)

[♡ Wishlist](#)

# Preset

- ◆ Si possono aggiungere nelle pagine:
  - ◆ homepage
  - ◆ shop
  - ◆ product
  - ◆ cart
  - ◆ blog
  - ◆ post
  - ◆ about-us, contact-us
  - ◆ page [I preset saranno uguali in tutte le custom pages](#)

# Preset

- ◆ Può essere inserito in ogni pagina N volte
- ◆ Ogni istanza ha i suoi valori nella variabile theme
- ◆ Può esserne limitato l'inserimento su alcune pagine

## Creare un preset

1. Definire un id univoco per il proprio preset
2. Stabilire un gruppo di appartenenza (content, media, products, ... )
3. Creare le configurazioni nel file preset.json
4. Creare un file **template\_ID\_PRESET.html**
5. Inserire nel file html i dati dinamici dell'istanza corrente del preset

# preset.json

```
{  
  "placeholders" : {  
    "image": "https://static-cdn.storen.com/themes/config\_v2/svg/image\_placeholder.svg",  
    ...  
  },  
  "presets" : {  
    "PRESET_ID" : {  
      ...  
    },  
    "PRESET_ID" : {  
      ...  
    },  
    ...  
  }  
}
```



Add new section

# Preset.json esempio paragraph

1. ID: paragraph: senza spazi, maiuscole o caratteri speciali/accentati. Usare \_
2. Campi

◆ title	→ type text	→ multilang
◆ content	→ type editor	→ multilang
◆ link	→ type link	→ multilang
◆ show_image	→ yes_no	→ no multilang
◆ image	→ type image	→ no multilang
◆ Image_alignment	→ select	→ no multilang

3. template\_paragraph.html

# Preset.json esempio paragraph: image

```
{
  "paragraph": {
    "group": "Content",
    "label": "Paragraph + image",
    "icon": "fa-align-left",
    "available": [],
    "settings": [
      { "label": "Text content", "fields": [ ... ] },
      { "label": "Image", "fields": [ ... ] } ,
    ]
  }
}
```

## < Paragraph and image

CONTENT



IMAGE





# Preset.json esempio paragraph: image

```
+file : template/template_paragraph.html
```

```
<section class="section template_paragraph" id="section_{{vars.section.id}}_paragraph">  
  <div class="{{vars.layout}}">  
    ...  
  </div>  
</section>
```

# Preset.json esempio paragraph: image

<code>{{vars.section.id}}</code>	id dell'istanza corrente.
<code>{{vars.layout}}</code>	passa il layout della pagina (homepage,shop,blog ... ) anche alle sezioni, può comunque essere sostituito da variabili del preset.
<code>{{vars.section.settings}}</code>	tutte le configurazioni salvate per l'istanza corrente.
<code>{{vars.section.data_editor}}</code>	da utilizzare per costruire gli attributi data-editor per i campi con live editing.

Vedere come esempio il file **homepage.html** per vedere come è composta la variabile **vars**

# Preset.json esempio paragraph: image

```
"settings": [
  { "label": "Text content", "fields": [ ... ] },
  {
    "label": "Image",
    "fields": [
      {
        "type": "image",
        "id": "image",
        "label": "Image",
        "multilang": "no",
        "info": "Free dimension",
        "default": ""
      }
    ]
  }
],
```

## Image ⓘ



Click here to upload a file or select an image from your gallery

## Alternate text ⓘ

# Preset.json esempio paragraph: image

Il campo image genera automaticamente un campo ALT, che diventa multilingua se lo è anche l'immagine.

## Se non multilingua:

```
vars.section.settings.image  
vars.section.settings.image_ALT
```

## Se multilingua:

```
attribute(vars.section.settings,"image_~locale)  
attribute(vars.section.settings,"image_ALT_~locale)
```

Anche il data-editor va declinato se il campo è multilingua:

```
{{vars.section.data_editor}}_image_{{locale}}
```

## Image ⓘ



Click here to upload a file or select an image from your gallery

## Alternate text ⓘ

# Preset.json esempio paragraph: image

## Non multilingua

```

```

## Multilingua

```

```

# Filter

## FILTER

Filtro e opzione filtro (\_INDEX)

Filter to use

Filter value ⓘ

```
{% set filterName = 'parameters.' ~ vars.section.settings.filter %}  
{% set filterValue = vars.section.settings.filter_INDEX %}  
  
{% set filterArray = [] %}  
{% set filterArray = filterArray | merge({ (filterName) : (filterValue) }) %}  
{% set sortArray = [] %}  
{% set items = getItemsByFilter(null, null, filterArray, sortArray, 0, vars.section.settings.count) %}
```

<https://themes-developers.store/en.com/hc/it/articles/360019514252-Prodotti>

# Pixel/Number

## PIXEL

```
body{ font-size: {{theme.settings.general.font.body_font_size}}px }
```

### Font size

px

## NUMBER

```
{% set items = getHighview() %}  
{% set items = items | slice(0, vars.section.settings.count ) %}
```

### Max number of products

# Textarea/Editor

## TEXTAREA/EDITOR

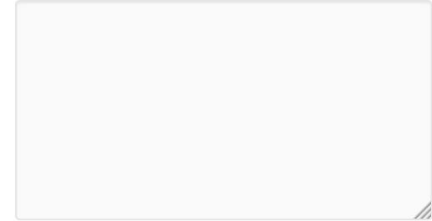
### Non multilingua

```
{% vars.section.settings.content | raw %}
```

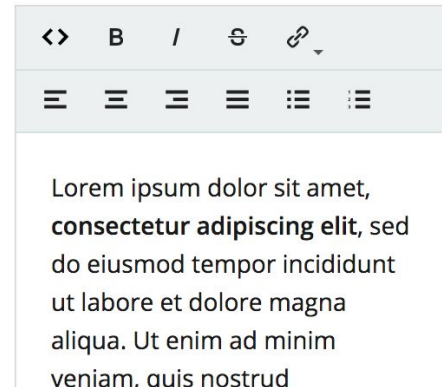
### Multilingua

```
{% attribute(vars.section.settings, "content_~locale ) | raw %}
```

### Map URL (Embed Iframe)

A large, empty rectangular box with a light gray border and a small diagonal slash in the bottom right corner, representing a placeholder for an embedded iframe.

### Content

A rich text editor interface. The top toolbar contains icons for source code (<>), bold (B), italic (/), strikethrough (ABC), and link (chain). Below the toolbar is a row of six list-style icons: bulleted, numbered, none, left-align, right-align, and centered-align. The main editing area contains the text: "Lorem ipsum dolor sit amet, **consectetur adipiscing elit**, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam. quis nostrud".

Lorem ipsum dolor sit amet, **consectetur adipiscing elit**, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam. quis nostrud



# Riassumendo..

## TIPI DI CONFIGURAZIONI

text, pixel, color, select, yes\_no, link, image, navigation, textarea, editor, filter, number, category, brand, blog\_category, gallery

## LIVE EDITING

text, color, textarea, editor

Disattivare live editing:

solo per text, image

**"light" : "no"**

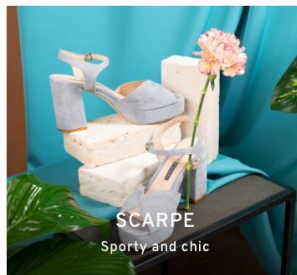
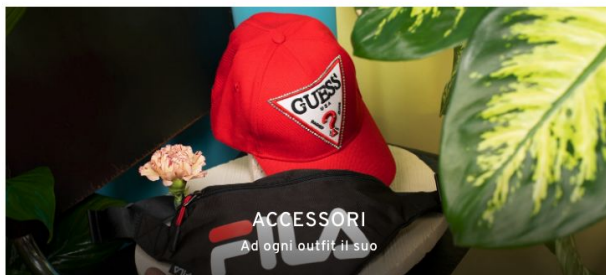
## MULTILANG

text, select, yes\_no, link, image, navigation, textarea, editor, category, blog\_category

Riepilogo aggiornato: <https://themes-developers.storenden.com/hc/it/articles/360019251792-Config-JSON-riepilogo>

# Esempi

<https://www.negozipellizzari.it/donna>



Widget con 4 elementi, ciascuno con

- ◆ Immagine
- ◆ Link
- ◆ Titolo
- ◆ Sottotitolo

E un insieme di configurazioni generali sul widget: colori, font, ...

# Esempi

<https://www.negozipellizzari.it/bambino>



Hero banner con due aree cliccabili, destra e sinistra.

- ◆ Immagine
- ◆ Titolo e configurazione stile del titolo
- ◆ Label e link dei pulsanti
- ◆ Stile dei pulsanti

# Esempi

<https://www.evergreenweb.it/>



## Materassi in memory foam

I materassi memory distribuiscono il peso in modo uniforme, assicurando in questo modo una piacevole sensazione di supporto su tutto il corpo.

Vedi tutti →



**DUAL SEASON**  
MEMORY E LATTICE AIR



+ 2 CUSCINI  
MEMORY  
GRATIS

Materasso in Lattice Air di origine NATURALE, Memory Foam e Waterfoam alto 20 cm

~~€ 725,00~~ € 259,00

← Scopri di più

## ENERGY ALMOND

MEMORY 2 STRATI



Materasso Memory Foam Ortopedico Sfoderabile 4 Lati Allergico H20cm | Energy

~~€ 359,00~~ € 177,00

← Scopri di più



**MARE PLUS 4 CM**  
MEMORY ONDULATO 2 STRATI



+ 2 CUSCINI  
MEMORY  
GRATIS

Materasso Memory Foam alto 22 cm Onda MASSAGGIANTE Ortopedico Fodera Antiacaro

~~€ 600,00~~ € 169,00

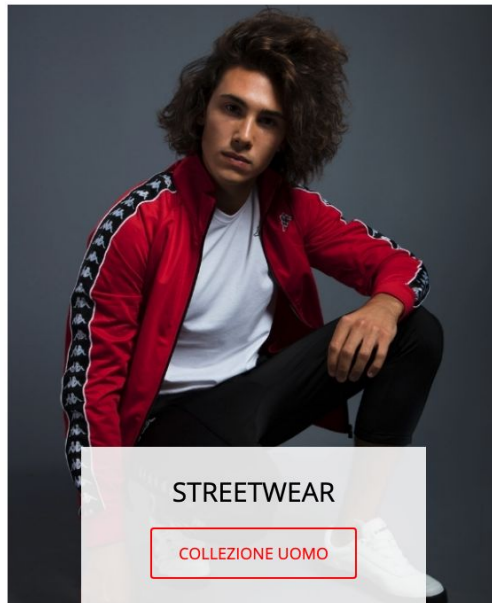
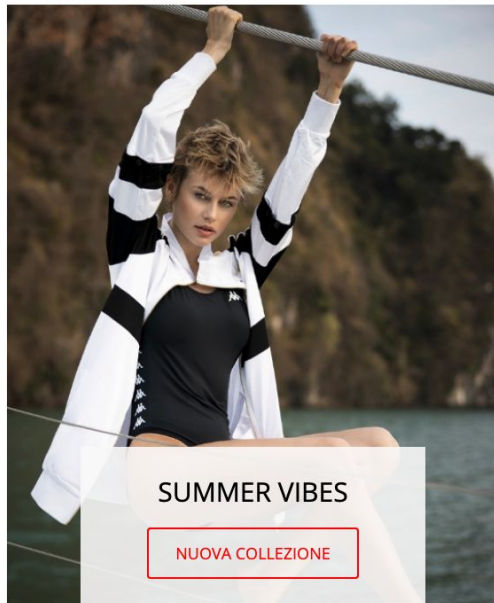
← Scopri di più

Widget prodotti in primo piano con

- ◆ Titolo
- ◆ Descrizione
- ◆ Icona
- ◆ Link
- ◆ Link label
- ◆ Colore del box laterale

# Esempi

<https://bidon1938.com/>



Widget con due elementi,  
ciascuno con

- ◆ Titolo
- ◆ Link
- ◆ Link label
- ◆ Immagine

# Esempi

<https://www.garmont.com/>

Widget selezione prodotti: l'utente inserisce i codici SKU dei prodotti che vuole far comparire nel carousel.

## OUR TOP PICKS



TOUBKAL GTX



TIKAL



PINNACLE GTX



AGAMURA

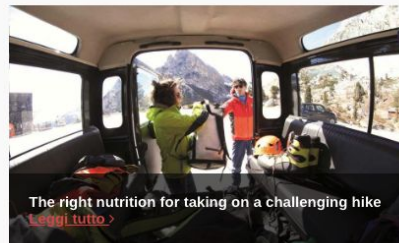
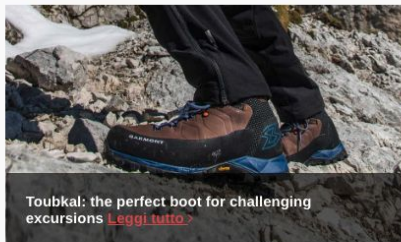


# Esempi

<https://www.garmont.com/>

L'utente può selezionare la categoria o il tag con cui filtrare i prodotti

## WILD STORIES



# Esempi



## RIBELLE TECH OD

Provide an ultra-light, high performance boot that's perfect for fast and light alpinism.

[VIEW PRODUCT](#)

Widget product glorification,  
un insieme di slide  
(carousel) composto da

- ◆ Titolo
- ◆ Descrizione
- ◆ Link
- ◆ Link label
- ◆ Immagine



# Esempi



## GRAPPERIA NARDINI

*Il fascino della tradizione*

La Grapperia sul Ponte è la sede storica della Nardini, la più antica distilleria d'Italia e l'unica originaria di Bassano. Un luogo dal grande fascino che dal 1779 ha visto susseguirsi i grandi cambiamenti della storia d'Europa.

Annoverata fra i Locali Storici d'Italia, la Grapperia, affacciata sul fiume Brenta, è rimasta immutata nel tempo e regala ancora oggi un'atmosfera unica. Un luogo trasversale e vivace dove bassanesi e turisti si incontrano per assaporare la magia di un prodotto, di una storia.

Widget con apertura/chiusura dettagli mediante jQuery.

- ◆ Titolo
- ◆ Colore e sfondo del titolo
- ◆ Sfondo e colori del widget
- ◆ Immagine
- ◆ Sotto titolo
- ◆ Testi
- ◆ N immagini che compongono il carousel immagini

4.

# CSS custom per widget

Per singola istanza

## CSS custom per i widget

Alcuni widget necessitano di stili CSS da applicare alla singola istanza.

Questi widget sono tipicamente widget usati più volte e che si differenziano ad esempio per stile.

## Esempio

Esempio: un widget con uno sfondo colorato che cambia a seconda di dove lo devo inserire, e per il quale è necessario che venga adattato anche il colore dei testi che contiene.

Un widget può essere inserito  $N$  volte, quindi non possiamo usare configurazioni globali.

# Esempio



## NEW PRODUCT

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

[BUY IT NOW](#)



## NEW in product

Lorem ipsum dolor sit amet, **consectetur adipiscing elit**, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

[Buy now!](#)

# Soluzione 1

Definire un insieme di regole generali e nei widget far selezionare mediante una select lo stile da applicare all'elemento.

→ va bene se conosco tutte le possibili combinazioni in fase di creazione del tema.



The image shows a mobile application interface for configuring button styles. The title is 'Buttons styles' with a back arrow. There are four sections, each representing a button style:

- BUTTON STYLE 1** (selected):
  - Font size: 15 px
  - Font weight: Black (900)
  - Color: #FFFFFF
  - Text color: #1F1F1F
  - Border color: #1F1F1F
  - Color on hover: #1F1F1F
  - Text color on hover: #FFFFFF
  - Border color on h...: #1F1F1F
- BUTTON STYLE 2**
- BUTTON STYLE 3**
- BUTTON STYLE 4**

# Soluzione 1

- A. Si definiscono le configurazioni in schema.json all'interno di "general"

```
{
  "label": "Buttons styles",
  "id": "buttons",
  "icon": "fa-square",
  "settings": [{
    "label": "Button style 1",
    "fields": [
      {
        "type": "pixel",
        "id": "font_size_1",
        "label": "Font size",
        "default": "15"
      },
      {
        "type": "select",
        "id": "font_weight_1",
        "label": "Font weight",
        "default": "400",
        "options": [ ... ]
      },
      {
        "type": "color",
        "id": "buttons_background_1",
        "label": "Color",
        "default": "#698EB8"
      },
      {
        "type": "color",
        "id": "button_textcolor_1",
        "label": "Text color",
        "default": "#FFF"
      },
      ...
    ]
  },
  {
    "label": "Button style 2",
    "fields": [ ... ]
  }
]
```

# Soluzione 1

- B. Si creano le classi CSS in theme.style che serviranno a rappresentare le combinazioni di stili

```
.button_2{
  outline: none;
  font-size: {{theme.settings.general.buttons.font_size_2}}px;
  font-weight: {{theme.settings.general.buttons.font_weight_2}};
  background: {{ theme_value(theme.settings.general.buttons.buttons_background_2, "settings.general.buttons.buttons_background_2") }};
  color: {{ theme_value(theme.settings.general.buttons.button_textcolor_2, "settings.general.colors.button_textcolor_2") }};
  border: 2px solid {{ theme_value(theme.settings.general.buttons.button_border_2, "settings.general.colors.button_border_2") }};
}

.button_2:hover, .button_2:focus, .button_2:active{
  box-shadow: 0 2px 4px rgba(0,0,0,0.2);
  -moz-box-shadow: 0 2px 4px rgba(0,0,0,0.2);
  -webkit-box-shadow: 0 2px 4px rgba(0,0,0,0.2);
  -o-box-shadow: 0 2px 4px rgba(0,0,0,0.2);
  -ms-box-shadow: 0 2px 4px rgba(0,0,0,0.2);
  background: {{ theme_value(theme.settings.general.buttons.buttons_background_2_hover, "settings.general.buttons.buttons_background_2_hover") }};
  color: {{ theme_value(theme.settings.general.buttons.button_textcolor_2_hover, "settings.general.colors.button_textcolor_2_hover") }} !important;
  border: 2px solid {{ theme_value(theme.settings.general.buttons.button_border_2_hover, "settings.general.colors.button_border_2_hover") }};
}
```



# Soluzione 1

- C. In preset.json si aggiunge una select con le classi come valori delle option da applicare agli elementi.

```
{
  "type": "select",
  "id": "button_style",
  "label": "Button style",
  "default": "button_1",
  "options": [
    {
      "option": "button_1",
      "label": "Button style 1"
    },
    {
      "option": "button_2",
      "label": "Button style 2"
    },
    {
      "option": "button_3",
      "label": "Button style 3"
    },
    {
      "option": "button_4",
      "label": "Button style 4"
    }
  ]
}
```

## ← Hero banner

LEFT LINK ▲

Button style

Button style 2 ▼

## Soluzione 2

Si definiscono i campi di configurazione direttamente in `preset.json`, e si usano nel `template .html` del widget.

## Soluzione 2

- A. Inserire le configurazioni direttamente nel file `preset.json` in corrispondenza della definizione del widget

→ i campi **color** non devono avere la proprietà `light`

```
{
  "type" : "color",
  "id" : "first_row_text_color",
  "light" : "no",
  "label" : "Text color",
  "default" : "#FFFFFF"
},
{
  "type" : "pixel",
  "id" : "first_row_font_size",
  "label" : "Font size",
  "default" : "12"
},
{
  "type" : "pixel",
  "id" : "first_row_line_height",
  "label" : "Line height",
  "default" : "23"
}
```

## Soluzione 2

- B. Inserire il tag `<style>` nel template html del widget e collegare gli stili

```
<style>

#ID dell'istanza >> .classe dell'elemento a cui si vuole applicare lo stile

#section_{{vars.section.id}}_hero_banner .first_row{
  font-size: {{vars.section.settings.first_row_font_size}}px;
  line-height: {{vars.section.settings.first_row_line_height}}px;
  font-weight: {{vars.section.settings.first_row_font_weight}};
  color: {{vars.section.settings.first_row_text_color}};
}

</style>
```

5.

# locale\_ISO.json

localizzazione

# Localizzazione

Oltre ai campi di testo configurabili, si possono modificare e aggiungere etichette del tema.

Nel tema, ogni etichetta è rappresentata con la funzione “lang”

```
{{ lang('page.home') }}           → “Homepage”  
{{ lang('page.all.items') }}      → “Tutti i prodotti”
```

Le etichette di sistema sono quasi interamente fornite insieme al tema per le lingue base del sistema.

# Localizzazione

E' possibile definire nuove etichette semplicemente dal menu

**Ecommerce > Tema > Traduzioni**

Cliccando sul pulsante “aggiungi traduzione”.

Chiave di traduzione

---

Aggiungi traduzione

Va inserita una nuova chiave, in caratteri minuscoli, senza spazi e con i punti se serve separare concettualmente le parole.

# Localizzazione

Una volta inserita, è possibile personalizzare il valore:

Chiave di traduzione	Seleziona: 
<code>this.is.a.new.label</code> <code>this.is.a.new.label</code>	<input data-bbox="643 546 1787 609" type="text" value="HELLO!"/> 

Dopodiché, è possibile utilizzarla nel file html del tema:

```
<span>
  {{ lang('this.is.a.new.label') }}
</span>
```



# Localizzazione

- ◆ Le etichette si memorizzano nei file lang\_ISO.json all'interno della directory locales del tema.
- ◆ Non si ereditano nei temi secondari. Ogni tema ha il suo set di etichette.
- ◆ Possono essere aggiunti nuovi file lingua, saranno funzionanti solo se la lingua è abilitata nel negozio.

6.

# storeden.js

Funzioni js di sistema

# storeden.js

- ◆ **Storeden.postProduct**(id,count,variants,callback)  
aggiunta di un prodotto al carrello
- ◆ **Storeden.checkShipping**(cartId, stateID, countryID, ,callback)  
ottenere le spedizioni disponibili
- ◆ **Storeden.login**(email, password, callback)  
login utente
- ◆ **Storeden.checkout**(...) e **Storeden.guestCheckout**(...)

[Vedi la guida completa sul file storeden.js](#)

7.

# Twg + Storeden

Funzioni utili

# Twig + Storeden

Funzioni per la rappresentazione dei dati Storeden nel frontend del sito.

[Lista di tutte le funzioni disponibili](#)

Le funzioni vengono aggiunte spesso e in base alle necessità, se non trovi la funzione che ti serve scrivi a [customer@storeden.com](mailto:customer@storeden.com) e valuteremo la possibilità di fornirti la funzione frontend che ti serve.

# Fine.

Grazie per l'attenzione. Domande?

[customer care@storeden.com](mailto:customer care@storeden.com)

