

# Ajouter des modules sur l'accueil d'E-GESCOM

EGescom => Accueil => Dashboard

- [Versions du document](#)
- [Description des étapes à réaliser](#)
- [Visualisation côté E-GESCOM](#)
- [Fonctionnement de la macro](#)

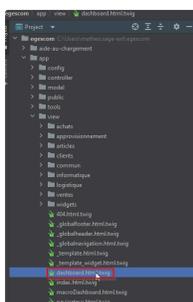
## Versions du document [↗](#)

Date	Auteur	Lien ticket Jira
12/06/2024	Matéo	<a href="https://proapro.atlassian.net/browse/GGW-99">https://proapro.atlassian.net/browse/GGW-99</a>

## Description des étapes à réaliser [↗](#)

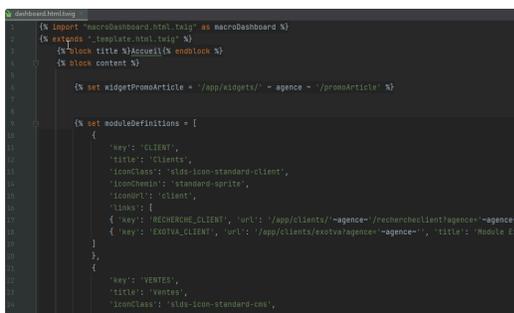
### 1. PhpStorm :

- Ouvrir l'application d'environnement de développement.
  - Choisir => egescom => app => view => dashboard.html.twig :



Doubles cliquez

- Nous voici dans le fichier dashboard :



### 2. Ajout d'un module : spécifications nécessaires pour la déclaration.

- Pour créer le module, il est nécessaire de spécifier les caractéristiques suivantes dans la déclaration "moduleDefinitions" :

```
{% set moduleDefinitions = [
  {
```

- La clé du module "key".
- Le titre du module "title".
- L'icone de la classe pour l'icone du module "iconClass".
- Chemin attribué pour l'icone d'un module "iconChemin".
- L'url de l'icone "iconUrl".
- Les liens pour les sous-modules "links".

- On y donne la clé "key" pour les sous-modules en y entrant les mêmes caractéristiques lors de l'initialisation du module : spécifications nécessaires pour la déclaration.

### 3. Déploiement des valeurs pour l'ajout.

```
{
  'key': '',
  'title': '',
  'iconClass': '',
  'iconChemin': '',
  'iconUrl': '',
  'links': [
    { 'key': '', 'url': '', 'title': '', 'icon': '', 'iconClass': '', 'iconChemin': '' }
  ]
}
```

Avant de donner les valeurs.

```
key: ARTICLE,
title: ARTICLE,
iconClass: 'fa fa-steam-rocket',
iconChemin: 'resources/icon',
iconUrl: 'product',
links: [
  { key: RECHERCHE_ARTICLE, url: '/app/article?agence=/resources/icon?agence=', title: 'Module Recherche Article', icon: search, iconClass: 'fa fa-steam-rocket', iconChemin: 'resources/icon' }
]
```

Après avoir donnée les valeurs

Nous avons plus qu'à faire un déploiement de notre ajout.



## Visualisation côté E-GESCOM [↗](#)

### 1. Egescom : Module vide vs Module rajouté

```
{
  'key': '',
  'title': '',
  'iconClass': '',
  'iconChemin': '',
  'iconUrl': '',
  'links': [
    { 'key': '', 'url': '', 'title': '', 'icon': '', 'iconClass': '', 'iconChemin': '' }
  ]
}
```

Si nous n'attribuons aucune valeur à nos variables dans notre nouveau module, nous constaterons qu'il est vide.

Description des variables :

- 'key': '' => Représente le Module principal
- 'title': '' => Représente le nom du Module principal
- 'iconClass': '' => Représente la classe de l'icône assigné au Module Principal
- 'iconChemin': '' => Représente le chemin de l'icône du Module principal

```
<use xlink:href="/app/public/assets/icons/{{ iconChemin }}/svg/symbols.svg#{{ iconUrl }}"></use>
```

- 'iconUrl': '' => Représente l'icône dans l'url
- 'links': [ => Représente le lien des sous-modules avec le Module principal
  - 'key': '' => Représente le/les Sous-module(s)
  - 'title': '' => Représente le titre du Sous-module
  - 'url': '' => Représente le lien du Sous-module

```
< href={{ link.url }} role="menuitem" tabindex="1" style="font-size: 1.2em; padding: 5px 10px; border: 1px solid #ccc; border-radius: 4px; display: inline-block; text-decoration: none; color: #333; background-color: #fff; margin-right: 5px; vertical-align: middle;">
  <span class="list-item" title="{{ link.title }}">
    <svg class="list-item icon-small" style="float: right; margin-right: 5px; width: 1em; height: 1em; vertical-align: middle; border: 1px solid #ccc; border-radius: 4px; display: inline-block; text-decoration: none; color: #333; background-color: #fff; margin-right: 5px; vertical-align: middle;"/>
    </span>
  </span>
</a>
```

- 'iconClass': '' => Représente l'icône du Sous-module
- 'iconChemin': '' => Représente le chemin de l'icône du Sous-module

]



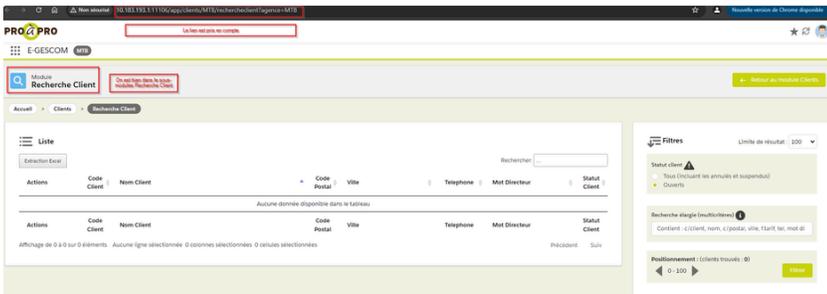
Ci-dessus, le module que nous avons ajouté n'est pas visible.

```
key: ARTICLE,
title: ARTICLE,
iconClass: 'fa fa-steam-rocket',
iconChemin: 'resources/icon',
iconUrl: 'product',
links: [
  { key: RECHERCHE_ARTICLE, url: '/app/article?agence=/resources/icon?agence=', title: 'Module Recherche Article', icon: search, iconClass: 'fa fa-steam-rocket', iconChemin: 'resources/icon' }
]
```

En assignant des valeurs aux variables, le résultat obtenu est le suivant :



On y voit que le module a été rajouté



## Fonctionnement de la macro [↗](#)

1. Voici ce qui nous permet d'ajouter notre nouveau module sur le dashboard:

- C'est une macro Twig :

```

1 {% macro dashboard(modules, moduleKey, moduleTitle, iconClass, moduleLinks, session, iconUrl, iconChemin, agence) %}
2     {% if attribute(modules, moduleKey)[0] %}
3         <div class="slds-col slds-size_1-of-1 slds-large-size_1-of-3 slds-m-bottom_medium">
4             <div class="slds-box">
5                 <article class="slds-card">
6                     <div class="slds-card_header slds-grid">
7                         <header class="slds-media slds-media_center slds-has-flexi-truncate">
8                             <div class="slds-media_figure">
9                                 <span class="slds-icon_container {{ iconClass }}" title="Module {{ moduleTitle }}">
10                                    <svg class="slds-icon slds-icon_small" aria-hidden="true">
11                                        <use xlink:href="/app/public/assets/icons/{{ iconChemin }}/svg/symbols.svg#{{ iconUrl }}"></use>
12                                    </svg>
13                                    <span class="slds-assistive-text">Module {{ moduleTitle }}</span>
14                                </span>
15                            </div>
16                            <div class="slds-media_body">
17                                <h2 class="slds-card_header-title">
18                                    <a href="/app/{{ moduleTitle|lower }}?agence={{ agence }}" class="slds-card_header-link slds-truncate" title="Module {{ moduleTitle }}">
19                                        <span>Module {{ moduleTitle }}</span>
20                                    </a>
21                                </h2>
22                            </div>
23                            <div class="slds-no-flex">
24                                <button class="slds-button slds-button_icon slds-button_icon-border-filled" id="bt-toogle-{{ moduleKey|lower }}">
25                                    <svg class="slds-button_icon" aria-hidden="true" id="icon-toogle-{{ moduleKey|lower }}">
26                                        <use xlink:href="/app/public/assets/icons/utility-sprite/svg/symbols.svg#chevrondown"></use>
27                                    </svg>
28                                    <span class="slds-assistive-text">Open</span>
29                                </button>
30                            </div>
31                        </header>
32                    </div>
33                    <div class="slds-card_body slds-card_body_inner slds-is-collapsed" id="content-toogle-{{ moduleKey|lower }}">
34                        <ul class="slds-has-dividers_top slds-has-block-links_space">
35                            {% for link in moduleLinks %}
36                                {% if attribute(modules, moduleKey)[1][link.key][0] %}
37                                    <li class="slds-item">
38                                        <a href="{{ link.url }}" role="menuitem" tabindex="-1" style="color:black;">
39                                            <span class="slds-truncate" title="{{ link.title }}">
40                                                <svg class="slds-icon slds-icon_small {{ link.iconClass }}" slds-m-right_small" aria-hidden="true">

```

```

41         <use xlink:href="/app/public/assets/icons/{{ link.iconChemin }}/svg/symbols.svg#{{ link.icon }}"></use>
42     </svg>{{ link.title }}
43 </span>
44 </a>
45 </li>
46 {% endif %}
47 {% endfor %}
48 </ul>
49 </div>
50 </article>
51 </div>
52 </div>
53 {% endif %}
54 {% endmacro %}

```

Dans le fichier principal "dashboard.html.twig", vous trouverez les variables ajoutées.

## 2. Comment faire pour l'ajouter sur notre Template ?

- On importe la macro Twig dans la Template principale en y mettant un alias :

```

dashboard.html.twig  macroDashboard.html.twig
{% import 'macroDashboard.html.twig' as macroDashboard %}

```

- Ensuite nous appelons la macro :

```

appel macro Dashboard (ajouts modules)#
<div class="slds-grid slds-wrap slds-gutters">
  <div class="slds-col slds-size_1-of-1 slds-large-size_12-of-12 slds-m-bottom_medium">
    <div>
      <div class="slds-card">
        <div class="slds-p-around_medium">
          <div class="slds-grid slds-wrap slds-gutters">
            {% for module in moduleDefinitions %}
              {{ macroDashboard.modules, module.key, module.title, module.iconClass, module.links, session, module.iconUrl, module.iconChemin, agence }}
            {% endfor %}
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

Voilà la macro dans une boucle for avec les données de chaque colonne. On voit les noms des modules, le titre de chaque module, l'import de la classe, et le nom de la macro. Dans la première case, on voit les mêmes variables que la macro.