

Intégrations graphiques Publik - Development #66786

migrer les styles layout de gru-content vers une nouvelle class

29 juin 2022 17:05 - Thomas Jund

Statut:	Fermé	Début:	29 juin 2022
Priorité:	Normal	Echéance:	
Assigné à:	Thomas Jund	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
la class .gru-content a un double rôle :			
<ul style="list-style-type: none">• Limiter le style des cellules, boutons, grille float et autre à son scope.• Permettre le layout sidebar + colonne(s).			
Ce double rôle pose problème. Le principal cas posant problème est la difficulté de proposer des placeholders combo dans les `{% block content-pre %}` et `{% block content-post %}`. Les cellules ajoutés au sein de ces blocks ne seront pas stylés.			
Pour permettre d'hériter des styles définis pour les cellules dans ces blocks, 2 solutions sont possibles :			
<ol style="list-style-type: none">1. agrandir le scope du style des cellules de `.gru-content` à `main`.2. Pouvoir ajouter la class .gru-content à d'autres blocks pour initier optionnellement un scope "cellules stylées" à d'autres éléments du template.			
<ol style="list-style-type: none">1. est difficile car cela va engendrer des effets de bords sur tous les thèmes qui ont personnalisés des cellules hors .gru-content.2. Est pour le moment non possible car .gru-content va en plus apporter un layout (son 2e rôle) qu'on ne veut clairement pas.			
Ce ticket pour proposer de faire supporter l'initiation du layout sidebar + columns à une autre class et ainsi pouvoir multiplier les .gru-content.			
Demandes liées:			
Lié à Intégrations graphiques Publik - Development #66170: Thème Nantes		Fermé	13 juin 2022
Lié à Intégrations graphiques Publik - Development #67127: CSS: supprimer la ...		Fermé	07 juillet 2022

Révisions associées

Révision c2fed953 - 08 juillet 2022 11:28 - Thomas Jund

theme: move .gru-content layout styles to new .central-content class (#66786)

Historique

#1 - 29 juin 2022 17:16 - Thomas Jund

- Fichier 0001-css-move-.gru-content-layout-styles-to-new-.default-.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Patch qui propose la class `.default-content`

#2 - 29 juin 2022 18:05 - Thomas Jund

je supprime également toute relation entre .gru-content et la balise .div. Ouvrant la possibilité d'appliquer la class sur d'autres tags.

#3 - 29 juin 2022 18:06 - Thomas Jund

- Fichier 0001-css-move-.gru-content-layout-styles-to-new-.default-.patch ajouté

#4 - 29 juin 2022 18:07 - Thomas Jund

- Fichier 0001-css-move-.gru-content-layout-styles-to-new-.default-.patch supprimé

#5 - 29 juin 2022 18:07 - Thomas Jund

- Fichier 0001-css-move-gru-content-layout-styles-to-new-default-patch supprimé

#6 - 29 juin 2022 18:07 - Thomas Jund

- Fichier 0001-css-move-gru-content-layout-styles-to-new-default-patch ajouté

#7 - 29 juin 2022 18:16 - Frédéric Péters

la class .gru-content a un double rôle :

Pour info/mémoire, la raison historique n'est ni l'un ni l'autre mais simplement avoir une classe spécifique à la GRU, permettant de cibler nos éléments, quand on avait des intégrations graphiques où le contenu Publik était intégré dans de l'HTML/CSS existant.

#8 - 05 juillet 2022 16:19 - Thomas Jund

- Lié à Development #66170: Thème Nantes ajouté

#9 - 05 juillet 2022 16:29 - Serghei Mihai

- Statut changé de Solution proposée à Solution validée

#10 - 05 juillet 2022 16:29 - Frédéric Péters

- Statut changé de Solution validée à Solution proposée

#11 - 05 juillet 2022 16:30 - Frédéric Péters

Non, pas default-content, qui ne correspond à aucune convention de nommage (pas qu'on soit parfait là-dessus, je sais).

"gru" évoquait au moins la GRU.

"default" évoque un contenu par défaut mais ça n'est pas ça.

#12 - 06 juillet 2022 09:51 - Thomas Jund

Pour réexpliquer le but du patch :

le nom de la nouvelle class est pour celle qui initie le layout flex.

c'est pour le wrapper du layout "sidebar -- columns".

La class 'gru-content' devient la class qui permet aux cellules, boutons, d'hériter des styles définies pour eux.

donc en étant descriptif, on pourrait utiliser

```
`sidebar-columns-wrapper`
```

Autre idées bienvenues.

#13 - 06 juillet 2022 10:10 - Frédéric Péters

Pour réexpliquer le but du patch :

Mmm. "La class 'gru-content' devient la class qui permet aux cellules, boutons, d'hériter des styles définies pour eux." ça laisserait penser que cette classe serait ajoutée à d'autres endroits, non, ou je ne comprends pas ?

..

Mais sinon j'ai bien compris qu'on cherche un nom pour la zone centrale de la page, entre la navigation et le pied de page, mais sans la zone "messages". Je m'interrogerais d'ailleurs bien sur la position de celle-ci, mais laissons ça ailleurs, par contre ça me ferait quand même dès maintenant éloigner sidebar-columns-wrapper qui s'attache trop à la situation présente.

Comme on a déjà "main-content" pour la zone qui contient le tout (messages + ce dont on parle), qu'on ne va pas mettre very-main-content, peut-être quelque chose comme central-content ? (qui peut cependant laisser penser que la barre latérale est exclue).

#14 - 06 juillet 2022 10:34 - Thomas Jund

Mmm. "La class 'gru-content' devient la class qui permet aux cellules, boutons, d'hériter des styles définis pour eux." ça laisserait penser que cette classe serait ajoutée à d'autres endroits, non, ou je ne comprends pas ?

pourrait être ajoutés. Oui dans un premier temps.

Ce que j'explique dans la description du ticket :

Ce ticket pour proposer de faire supporter l'initiation du layout sidebar + columns à une autre class et ainsi pouvoir multiplier les .gru-content.

mais sans la zone "messages"

Et sans tous les placeholders combo que l'on souhaite positionner en dessous ou au dessus.

Le cas d'usage (toujours dans la description du ticket) :

Le principal cas posant problème est la difficulté de proposer des placeholders combo dans les `{% block content-pre %}` et `{ block content-post %}`. Les cellules ajoutées au sein de ces blocks ne seront pas stylées.

Comme on a déjà "main-content" pour la zone qui contient le tout (messages + ce dont on parle), qu'on ne va pas mettre very-main-content

Voilà pourquoi j'ai décliné en `défaut-content` pour indiquer "la zone de contenu disponible par défaut" dans le thème.

central-content

Je ne sais pas si c'est mieux. Mais ça me va (on pourra sans problème amender par la suite).

#15 - 06 juillet 2022 11:00 - Frédéric Péters

"Pour réexpliquer le but du patch" donc je ne renvoyais pas à la description du ticket etc.

Mais sinon j'ai bien compris qu'on cherche un nom pour la zone centrale de la page, entre la navigation et le pied de page, mais sans la zone "messages".

Et sans tous les placeholders combo que l'on souhaite positionner en dessous ou au dessus.

Mmm, dans le patch,

```
<div id="content" class="default-content gru-content">
  {% block content-top %}{% endblock %}
  {% block content %}
```

et on a une série d'intégrations qui font

```
{% block content-top %}
<div id="theme-top-section">
  {% placeholder "top-section" name="Haut de page" %}
</div>
{% endblock %}
```

et donc un placeholder combo qu'on souhaite positionner au-dessus, il se trouve dans le <div nouvelle-classe>.

Et de manière plus générale, on a dans templates/combo/page_template.html

```
{% block content %}
  {% block combo-content %}
    {% block page-content %}
      {% block sidebar %}{% endblock %}
      <div id="columns-wrapper">
        {% block columns-top %}
          {% placeholder "columns-top" name=_('Top of content') optional=True acquired=False outer_tag=True %}
          {% endblock %}
        ...
        {% block columns-bottom %}
          {% placeholder "columns-bottom" name=_('Bottom of content') optional=True acquired=False outer_tag=True %}
        ...
      </div>
    {% endblock %}
  {% endblock %}
{% endblock %}
```

et donc les placeholders "top of content" et "bottom of content" ils sont aussi dans la zone dont on parle.

Je ne vois ainsi pas à quoi tu fais référence avec "sans tous les placeholders combo que l'on souhaite positionner en dessous ou au dessus".

--

Je me dis qu'après ça on gagnerait à se faire quelques svg? schématiques de la structure des pages, ça servirait aussi à formaliser les sélecteurs qu'on mettrait un peu en avant, vs certains possibles mais plus occasionnels, ex pour 1 colonne et barre latérale,

```
#page
|-----|
| .site-header
| #header-wrapper
|-----|
| .site-nav
| #nav-wrapper
|-----|
| #main-content-wrapper
| #messages
| #content (.gru-content)
| #sidebar | #columns-wrapper
| <place- | <placeholder columns-top>
| holder | #columns
| sidebar> | <placeholder content>
| | <placeholder columns-bottom>
|-----|
| <footer> sans classe
| #footer-wrapper
| <placeholder footer>
|-----|
```

~

Pour dire que le patch actuel, le nom "default" me convient bien peu, que juste changé en "central" ça me va mieux, mais que sur le fond je ne capte pas vraiment où on va.

#16 - 06 juillet 2022 13:46 - Thomas Jund

mais que sur le fond je ne capte pas vraiment où on va

Ok, je reprends depuis le début en essayant de détailler plus à l'aide de schémas le besoin, la problématique.

Je pars du besoin de pouvoir ajouter des placeholders combo pour un contenu au-dessus et/ou en dessous de 'sidebar + columns' :

```
+-----+
|
|Top Content
|-----+
|+-----+ +-----+
||      | |      |
||  Sidebar | | Columns |
||      | |      |
||      | |      |
||      | |      |
|+-----+ +-----+
|-----+
|Bottom Content
|-----+
+-----+
```

Je vais prendre l'exemple de l'ajout d'un top-content avec le code actuel, mais la problématique est la même pour bottom-content.

Solution 1.

Ajouter le placeholder top-content au {% block content-top %} qui se trouve avant la sidebar et au sein de .gru-content. Comme .gru-content initie le layout .flex pour placer la sidebar à côté des colonnes, si j'ajoute un div à cet endroit je vais avoir

```
+-----+
|gru-content
|+-----+ +-----+ +-----+
||      | |      | |      |
||top content| | Sidebar | |Columns |
||      | |      | |      |
||      | |      | |      |
|+-----+ +-----+ +-----+
+-----+
```

Et il va falloir bidouiller du CSS à coup de `flex-wrap` et `flex-basis` dans le CSS du thème pour repositionner `top content` correctement. J'ai déjà testé cette solution (exemple que tu as mentionné), et je la trouve inutilement complexe. Par contre, les cellules sont stylées.

Solution 2

Ajouter le placeholder `top-content` au `{% block content-pre %}` qui se trouve avant `.gru-content`.

```
+-----+
|top Content |
+-----+
|gru-content |
|+-----++-----+ |
||Sidebar || Columns | |
||          ||          | |
||          ||          | |
|+-----++-----+ |
+-----+-----+
```

Dans ce cas, le layout n'est pas contraint par le contexte flex, il se place au-dessus, dans le flux. Mais les cellules, boutons posés dans `top_content` ne seront pas stylés.

Ce qui demande à nouveau des bidouilles CSS à ralonge. Cette solution se retrouve même pire que la première s'il est nécessaire que les cellules ou les boutons de top-content héritent des styles définis pour eux (et je ne parle pas de la grille float). Voilà pourquoi j'ai une préférence à utiliser la solution 1.

Le patch que je propose est donc de pouvoir utiliser la solution 2, sans casser l'existant : pouvoir ajouter un combo placeholder au sein de `{% block content-pre %}` et lui ajouter une class `.gru-content` dans le thème dans le cas où on souhaite hériter des styles. Aucune surcouche CSS nécessaire.

#17 - 06 juillet 2022 14:22 - Frédéric Péters

Le patch que je propose est donc de pouvoir utiliser la solution 2

Mais ça ne correspond pas à la disposition actuelle des placeholders et le patch ne modifie pas celle-ci. (?)

~

Si le problème est d'avoir display: flex sur .gru-content, pourquoi ne pas juste le retirer, le mettre sur #content ?

Des explications je me serais attendu à soit ça,

```
.gru-content {
-   display: flex;
...
+ #content {
+   display: flex;
```

soit ok si pas envie de l'id l'ajout d'une nouvelle classe,

```
.gru-content {
-   display: flex;
...
+ .nouvelle-classe {
+   display: flex;
```

+ templates/theme.html

```
- <div id="content" class="gru-content">
+ <div id="content" class="nouvelle-classe gru-content">
```

~

C'est semble-t-il ça sauf que ça n'a pas d'impact sur les placeholders qui sont à l'intérieur. Ce patch est-il un prélude à un autre qui déplacerait des placeholders ?

~

Aussi ça se mélange à des autres modifs qui rendent la compréhension confuse, genre premier chunk quel rapport ?

```
- div.gru-content div.cell div.cell-items-pagination {
+ .gru-content div.cell div.cell-items-pagination {
```

J'imagine volonté de nettoyer des sélecteurs, ça pourrait être un autre ticket pour clarifier les choses ?

#18 - 06 juillet 2022 15:52 - Thomas Jund

Mais ça ne correspond pas à la disposition actuelle des placeholders et le patch ne modifie pas celle-ci. (?)

Heu, si.

```
{% block content-pre %}{% endblock %}
<div id="content" class="default-content gru-content">
  {% block content-top %}{% endblock %}
  {% block content %}
  {% endblock %}
  {% block content-bottom %}{% endblock %}
</div> <!-- #content -->
{% block content-post %}{% endblock %}
```

Ça correspond. Inutile de déplacer des placeholders.

soit ok si pas envie de l'id l'ajout d'une nouvelle classe,

C'est exactement ce que propose le patch

```
diff --git a/static/includes/_layout.scss b/static/includes/_layout.scss
```

```
- .gru-content {
+ .default-content {
    display: flex;
    #sidebar {
      flex: 0 0 $sidebar-width;
@@ -131,7 +131,7 @@ div#main-content {
    width: auto;
  }
}
```

```
- .gru-content {
+ .default-content {
    flex-direction: column;
    #sidebar {
      flex: 0 0 auto;
    }
}
```

```
diff --git a/templates/theme.html b/templates/theme.html
```

```
{% block content-pre %}{% endblock %}
- <div id="content" class="gru-content">
+ <div id="content" class="default-content gru-content">
```

Aussi ça se mélange à des autres modifs qui rendent la compréhension confuse, genre premier chunk quel rapport ?

Pour ouvrir la possibilité d'associer le class .gru-content à d'autres sectionning tag : <https://dev.entrouvert.org/issues/66786#note-2>

header.gru-content

section.gru-content

Par exemple dans le thème de Nantes, j'utilise <header> :

```
{% block content-pre %}
  <header class="theme-page-header gru-content">
    {% placeholder "page_header" name="Entête de la page" optional=False acquired=False %}
  </header>
{% endblock %}
```

#19 - 06 juillet 2022 16:39 - Corentin Séchet

Mais ça ne correspond pas à la disposition actuelle des placeholders et le patch ne modifie pas celle-ci. (?)

Si je comprends bien, le bloc "top content" dans l'exemple de Thomas va dans {% block content-pre %}, pas dans {% block content-top %}

Pour dire que le patch actuel, le nom "default" me convient bien peu, que juste changé en "central" ça me va mieux, mais que sur le fond je ne capte pas vraiment où on va.

Si je comprends bien, encore une fois, ça pourrait être "gru-layout" ou "main-layout".

#20 - 06 juillet 2022 21:51 - Frédéric Péters

Mais ça ne correspond pas à la disposition actuelle des placeholders et le patch ne modifie pas celle-ci. (?)

Heu, si.

Une dernière fois, euh sûr non cf plus haut que ça n'était pas le cas général, cet extrait du code :

```
{% block content %}
  {% block combo-content %}
    {% block page-content %}
      {% block sidebar %}{% endblock %}
      <div id="columns-wrapper">
        {% block columns-top %}
          {% placeholder "columns-top" name=_('Top of content') optional=True acquired=False outer_tag=True %}
          {% endblock %}
        ...
        {% block columns-bottom %}
          {% placeholder "columns-bottom" name=_('Bottom of content') optional=True acquired=False outer_tag=True %}
        ...
      e %}
    {% endblock %}
```

alors je dois comprendre maintenant que quand j'écris "disposition actuelle des placeholders" tu entends non pas les {% placeholder %} présents actuellement mais la possibilité d'en déclarer un autre dans { block content-pre }{ endblock %}.

Pour reprendre mon schéma qui listait les placeholder existants, et noter ma compréhension :

```
#page
|-----|
| .site-header |
| #header-wrapper |
|-----|
| .site-nav |
| #nav-wrapper |
|-----|
| #main-content-wrapper |
| #messages |
>>>> on parle de changements pour un <placeholder> qui n'existe pas et viendrait ici, dans une intégration spécifique <<<<<
| #content (.gru-content)
| #sidebar | #columns-wrapper
| <place- | <placeholder columns-top>
| holder | #columns
| sidebar> | <placeholder content>
| | <placeholder columns-bottom>
>>>> ou un autre qui viendrait ici <<<<<
|-----|
| <footer> sans classe |
| #footer-wrapper |
| <placeholder footer> |
|-----|
```

Mais c'est concernant les placeholder actuels que j'avais ma question "Ce patch est-il un prélude à un autre qui déplacerait des placeholders ?". (et la réponse est "non aucun rapport on parle de placeholders qui n'existent pas de manière générale, juste un truc pour intégration spécifique"). (et ok mais je regrette qu'il n'y ait pas la perspective généraliste).

--

Pour reprendre plus haut, stopper les égarements, j'étais juste à trouver le nom default-content nul et pour alimenter la recherche d'un meilleur nom je notais que ça prenait les nouvelles zones ("placeholder") haut/bas de contenu ([#62413](#)) et c'est parti en sucette sur la réponse "Et sans tous les placeholders combo que l'on souhaite positionner en dessous ou au dessus" qui pour s'opposait à ce que je venais juste d'écrire.

--

Aussi ça se mélange à des autres modifs qui rendent la compréhension confuse, genre premier chunk quel rapport ?

Pour ouvrir la possibilité d'associer le class .gru-content à d'autres sectionning tag : <https://dev.entrouvert.org/issues/66786#note-2>

Ok premier chunk mais un autre,

```
- div.gru-content div#rub_service h3,
+ div#rub_service h3 {
```

quel rapport ?

~~

Sur le tout vraiment je pense ça aide d'avoir des commits isolés, qui s'expliquent indépendamment, 1/ avoir le display: flex sur une autre classe, 2/ permettre la classe gru-content sur quelque chose qui n'est pas un <div>, 3/ nettoyer des déclarations inutiles.

~~

Pour dire que le patch actuel, le nom "default" me convient bien peu, que juste changé en "central" ça me va mieux, mais que sur le fond je ne capte pas vraiment où on va.

Si je comprends bien, encore une fois, ça pourrait être "gru-layout" ou "main-layout".

Ça introduit ici une distinction entre ce que serait "layout" vs ce que serait "content" et sans explication je ne peux pas capter. (je découragerais l'introduction de ce nouveau découpage).

#21 - 07 juillet 2022 10:14 - Thomas Jund

quand j'écris "disposition actuelle des placeholders" tu entends non pas les {% placeholder } présents actuellement mais la possibilité d'en déclarer un autre dans { block content-pre }{ endblock %}.

Oui, sûrement parce que mon propos n'a jamais parlé d'autres chose que de cela.

Ce patch est-il un prélude à un autre qui déplacerait des placeholders ?

Maintenant je comprends ton interrogation. En tout cas ce patch facilitera la possibilité de proposer de nouveaux placeholders ou de déplacer des existants en dehors de #content. Si tu penses aux placeholders "{% placeholder "columns-top" name=_('Top of content') ...}" et "{% placeholder "columns-bottom" name=_('Bottom of content') ...}", je pense qu'ils ont leur raison d'exister ici, à leur place. Je pourrais leur reprocher de s'appeler `name=_('Top of content')` et `name=_('Bottom of content')` car ils se situent *après/à côté de la sidebar*. Mais pour moi c'est un autre débat, une suite possible à donner ce patch.

1/ avoir le display: flex sur une autre classe, 2/ permettre la classe gru-content sur quelque chose qui n'est pas un <div>, 3/ nettoyer des déclarations inutiles.

Ok, je découpe, je commit et on avance.

#22 - 07 juillet 2022 14:27 - Thomas Jund

- Lié à Development #67127: CSS: supprimer la relation entre .gru-content et la balise div ajouté

#23 - 07 juillet 2022 15:03 - Thomas Jund

- Fichier 0001-theme-move-gru-content-layout-styles-to-new-central-content.patch ajouté

#24 - 07 juillet 2022 17:44 - Thomas Jund

commits pour [#66786](#), [#67127](#) et [#67128](#) dispo dans l'ordre sur branch wip/66786-gru-content

#25 - 07 juillet 2022 18:18 - Frédéric Péters

- Statut changé de Solution proposée à Solution validée

#26 - 08 juillet 2022 11:31 - Thomas Jund

- Statut changé de Solution validée à Résolu (à déployer)

```
commit c2fed953bb1fe0979088b2d8bbe5df15865a4a56
Author: Thomas JUND <tjund@entrouvert.com>
Date: Thu Jul 7 14:49:05 2022 +0200
```

```
theme: move .gru-content layout styles to new .central-content class (#66786)
```

#27 - 08 juillet 2022 15:14 - Transition automatique

- Statut changé de Résolu (à déployer) à Solution déployée

#28 - 11 septembre 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

0001-css-move-gru-content-layout-styles-to-new-default.patch	3,02 ko	29 juin 2022	Thomas Jund
0001-theme-move-gru-content-layout-styles-to-new-centra.patch	1,29 ko	07 juillet 2022	Thomas Jund