

Gadjo - Development #63293

systeme d'onglets verticaux

29 mars 2022 11:29 - Frédéric Péters

Statut:	Fermé	Début:	29 mars 2022
Priorité:	Normal	Echéance:	
Assigné à:	Frédéric Péters	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Oui		
Description			
D'abord pour le paramétrage des cellules #62965 mais avec la perspective que ça puisse également servir ailleurs. (je pense notamment au paramétrage des champs).			
Demandes liées:			
Lié à Combo - Development #62965: revoir le style de paramétrage des cellules		Fermé	20 mars 2022

Révisions associées

Révision 71361429 - 01 avril 2022 10:44 - Frédéric Péters

general: add support for tabs (#63293)

Historique

#1 - 29 mars 2022 11:29 - Frédéric Péters

- Lié à Development #62965: revoir le style de paramétrage des cellules ajouté

#2 - 29 mars 2022 11:39 - Frédéric Péters

- Fichier 0001-general-add-support-for-tabs-63293.patch ajouté

- Fichier map-tabs.png ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

Code js et balisage via https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles/Tab_Role, seule modif la mise en accès global d'une fonction window.selectTab, utilisée pour permettre dans le js de la cellule de changer d'onglet actif en cas d'erreur dans un autre onglet.

Pour le balisage, on a donc :

```
<div class="pk-tabs-container">
  <div class="pk-tabs-container--tablist" role="tablist" aria-label="xxx">
    {% for tab in manager_tabs %}
      <button role="tab"
        aria-selected="{{ forloop.first|yesno:"true,false" }}"
        aria-controls="panel-XXX-{{ forloop.counter }}"
        id="tab-XXX-{{ forloop.counter }}"
        tabindex="{{ forloop.first|yesno:"0,-1" }}">{{ tab.name }}</button>
    {% endfor %}
  </div>
  <div class="pk-tabs-container--panels"> <!-- # ou <form -->
    {% for tab in manager_tabs %}
      <div id="panel-{{ cell.get_reference }}-{{ forloop.counter }}"
        role="tabpanel" tabindex="0" {% if not forloop.first %}hidden{% endif %}
        data-tab-slug="{{ tab.slug }}"
        aria-labelledby="tab-XXX-{{ forloop.counter }}">
        inclure contenu du panneau en question
      </div>
    {% endfor %}
  </div>
</div>
```

Côté rendu cf capture, c'est de [#62965](#) puis commentaires/discussions Stéphane et Pierre.

J'ai hésité à permettre une variante horizontale mais c'était de la distraction.

#3 - 29 mars 2022 12:34 - Thomas Jund

Le JS que tu as écrit est au format ES2015+

Non compris et non supporté par IE et certains navigateurs moins récents.

Ça va coincer tout le JS sur ces navigateurs.

Soit utiliser JS en ES5 et prototypage pour compatibilité (faisable), soit charger en module JS pour écrire du JS moderne et garder un fonctionnement dégradé mais fonctionnel pour les vieux browsers, ou fuck les vieux browsers sur le BO et gadjo, à débattre.

Sur le code actuel, il faut vérifier avec plusieurs cellules sur une même page, à la lecture je doute que ça fonctionne.

(Si on part sur du JS propre et moderne) je te conseil d'utiliser la syntaxe de class (écriture plus simple du prototypage) pour que chaque cell heritent leurs méthodes et objects. Aucun risque de conflit et on peut exposer les elements actifs et piloter de l'extérieur au besoin via une api.

Associer les methodes à un objet "pk_tabs" plutôt qu'à window.

Éviter les target.parentNode dans les méthodes.

Et nomenclature css, je propose de pourquoi pas simplifier le non du composant en "pk-tabs" simplement

```
.pk-tabs  
.pk-tabs--container (si besoin)  
.pk-tabs--tab-list  
.pk-tabs--tab  
.tk-tabs--panel-list  
.tk-tabs--panel
```

Sinon, je suis heureux de lire du JS sans jQuery :)

#4 - 29 mars 2022 13:50 - Frédéric Péters

- Fichier 0001-general-add-support-for-tabs-63293.patch ajouté

Non compris et non supporté par IE et certains navigateurs moins récents.

On parle de backoffice, on y exigeait déjà un navigateur convenable, mais on trouve gadjo.js employé en front pour l'affichage de popup de facture, ça serait dommage là que le js ne charge pas pour ce qui y serait vu comme une erreur de syntaxe. De ce que je vois il s'agirait de la notation => pour les fonctions + l'interpolation de chaine via `\${...}`.

Sur le code actuel, il faut vérifier avec plusieurs cellules sur une même page, à la lecture je doute que ça fonctionne.

J'ai plein de cellules sur la même page, en pratique ça fonctionne donc.

(Si on part sur du JS propre et moderne) je te conseil d'utiliser la syntaxe de class (écriture plus simple du prototypage) pour que chaque cell heritent leurs méthodes et objects. Aucun risque de conflit et on peut exposer les elements actifs et piloter de l'extérieur au besoin via une api. Associer les methodes à un objet "pk_tabs" plutôt qu'à window.

Il y a une seule méthode associée mais go pour créer un objet global pk_tabs et lui associer et qu'il soit accédé depuis le code combo.

Éviter les target.parentNode dans les méthodes.

L'idée était de prendre le code fourni sur mdn et qui suivait à mon sens certains bons standards; je suis vraiment plus à l'aise à ne pas triturer leur code.

~~

Patch attaché qui ajoute le pk_tabs global et sa méthode selectTab + fonction() plutôt que => + qui crée le sélecteur avec une concaténation classique + qui renomme les classes pour avoir pk-tabs et pk-tabs--tab-list et pk-tabs--container.

#5 - 29 mars 2022 14:44 - Thomas Jund

on y exigeait déjà un navigateur convenable

Ok, bon à savoir.

De ce que je vois il s'agirait de la notation => pour les fonctions + l'interpolation de chaine via `\${...}`

Je relève aussi la methode forEach sur les nodeList

J'ai plein de cellules sur la même page, en pratique ça fonctionne donc.

J'ai pas testé, mais ça me semble étonnant à cause de

```
const tabList = document.querySelector('.pk-tabs [role="tablist"]');
```

qui va sélectionner le premier nœud trouvé dans le code. Donc l'event keydown `tabList.addEventListener('keydown')` ne devrait être ajouté qu'à 1 seul element (et tu dois avoir 1 tablist par cellule)

L'idée était de prendre le code fourni sur mdn et qui suivait à mon sens certains bons standards;

Mouai, par exemple c'est recommandé d'utiliser `.key` (<https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/key>) plutôt que `.keyCode` (qui est déprécié <https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent/keyCode>). Je pense que d'un point de vue a11y ça reste un exemple valable, mais pas au niveau pattern.

Bon, je vais installer les patchs pour tester réellement et te faire une proposition.

#6 - 29 mars 2022 14:51 - Frédéric Péters

Je relève aussi la methode `forEach` sur les `nodeList`

Ok mais là pas de la syntaxe, donc pas de problème pour le front qui ne passera pas par là.

J'ai pas testé, mais ça me semble étonnant à cause de

Ah je comprends avec ce que tu pointes, c'est la navigation clavier, qui passe à travers tous les onglets de la page; je veux bien ton patch revu pour reprendre ça et le style que tu souhaiterais.

#7 - 31 mars 2022 13:29 - Frédéric Péters

- Fichier `0001-general-add-support-for-tabs-63293.patch` ajouté

Ah je comprends avec ce que tu pointes, c'est la navigation clavier, qui passe à travers tous les onglets de la page; je veux bien ton patch revu pour reprendre ça et le style que tu souhaiterais.

En attendant si on préfère voici une version du patch qui retire simplement la navigation clavier.

Avec la navigation clavier mais un bug, ou sans la navigation clavier, ça serait vraiment utile de ne pas bloquer ici [#62965](#).

#8 - 31 mars 2022 16:34 - Thomas Jund

J'arrive

#9 - 31 mars 2022 17:42 - Thomas Jund

- Fichier `0001-general-add-support-for-tabs-63293.patch` ajouté

Bon J'ai pas réussi à installer le patchs de Fred, git am qui chouine.
Donc pas de diff, désolé.

Côté CSS, le seul changement est l'ajout de `:focus-visible` sur les boutons pour avoir un feedback visible pour la navigation au clavier.

Côté JS :

- je suis passer en pattern propotype pour séparer proprement les methodes et elements de chaque tabs et éviter au max les parsing du DOM.
- J'ai modifié les touches de clavier par up et down, parce que nos tabs sont positionnés à la vertical (et pas à l'horizontale comme sur l'exemple MDN)
- J'ai ajouté l'objet `Tabs` à l'objet `gadjo_js` parce qu'il est là et qu'il ne sert à rien pour le moment, autant l'utiliser.

Sinon, une proposition à débattre. Etant donné qu'on a besoin de JS pour switcher les valeurs des attributs `arias`, `hidden` et autre, pourquoi ne pas laisser JS ajouter ces attributs aux éléments ? Je vois plusieurs avantages

- Ça évite la création de templates django avec des `{{ forloop.first|yesno:"true,false" }}`
- Ça permet d'avoir un formulaire qui fonctionne sans JS (et je trouve que c'est quand même une bonne pratique de tendre vers ça)

#10 - 01 avril 2022 10:44 - Frédéric Péters

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

Je n'ai pas trouvé comment faire pour appeler selectTab depuis l'extérieur,

- dans mon patch c'était pk_tabs.selectTab mais maintenant gadjo_js.Tabs c'est une fonction,
- dans le vide viser la fonction en allant gadjo_js.Tabs.prototype.selectTab ça ne marche pas il faut un objet,
- modifier le new gadjo_js.Tabs(el); pour stocker l'objet dans l'élément, pour avoir un objet sur lequel appeler,

```
-     new gadjo_js.Tabs(el);
+     el.tabs = new gadjo_js.Tabs(el);
```

puis côté combo,

```
-     pk_tabs.selectTab($stab_button[0]);
+     $current_tab.closest('.pk-tabs')[0].tabs.selectTab($stab_button[0]);
```

ça ne marche parce que selectTab obtient désormais le bouton via btn = e.target plutôt que directement en paramètre.

donc j'ajoute aussi,

```
-     const btn = e.target;
+     const btn = e.target && e.target || e;
```

Et j'ai poussé ça ainsi, on reviendra dessus si nécessaire,

```
commit 71361429a0d5a7141efb544236efe95760679d3e
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Sat Mar 26 17:30:03 2022 +0100
```

```
general: add support for tabs (#63293)
```

#11 - 01 avril 2022 12:17 - Transition automatique

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

#12 - 05 juin 2022 04:42 - Transition automatique

Automatic expiration

Fichiers

0001-general-add-support-for-tabs-63293.patch	3,82 ko	29 mars 2022	Frédéric Péters
map-tabs.png	24,6 ko	29 mars 2022	Frédéric Péters
0001-general-add-support-for-tabs-63293.patch	3,99 ko	29 mars 2022	Frédéric Péters
0001-general-add-support-for-tabs-63293.patch	3,17 ko	31 mars 2022	Frédéric Péters
0001-general-add-support-for-tabs-63293.patch	4,2 ko	31 mars 2022	Thomas Jund