

Publik - Development #54115

Wysiwyg Publik

19 mai 2021 10:58 - Thomas Jund (congé, retour le 29/04)

Statut:	Fermé	Début:	19 mai 2021
Priorité:	Normal	Echéance:	
Assigné à:	Benjamin Dauvergne	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Club:	Non
Patch proposé:	Non		
Planning:	Non		

Description

Ticket général de discussion autour de la mise en place progressif d'un remplaçant à CKEditor.

Présentation faite au club et financée

Ce développement doit permettre d'utiliser une interface d'édition « riche » dans plus d'endroits, particulièrement pour permettre une interface de saisie de commentaires « enrichis » dans les échanges entre usagers et agents via l'action Commentaire, où les contraintes d'accessibilité et sécurité sont importantes, et où on ne veut pas surcharger l'utilisateur d'une interface type traitement de texte.

Cette nouvelle interface sera ensuite étendue à d'autres usages où un contrôle est important, particulièrement pour la rédaction d'emails enrichis, où il faut composer avec les prises en charge variables des lecteurs de mails.

Pour une illustration de ce qui est imaginé :

<https://perso.entrouvert.org/~fred/tmp/our-editor.webm>

Cette version concernerait les éléments suivants :

- Paragraphes
- Puces
- Mise en forme basique (gras, italique)
- liens

Attention pas d'images à l'heure actuelle, cela pourra faire l'objet d'une proposition ultérieure.

Premières notes de cadrage

- En ligne de mire: que cela devienne notre éditeur wysiwyg (en remplacement de ckEditor).
- Pouvoir personnaliser les options de l'éditeur, parce qu'il ne faut pas les mêmes options pour rédiger un mail qu'un commentaire. Comme
 - proposer les balises inline uniquement
 - exclure certains niveaux de titres (comme commencer par h3).
 - rendre disponible les class .pk-
 - pouvoir proposer un module "grid"
 - accès aux ressources combo
 - proposer un block cartographie
 - possibilité de coder des modules sur mesure pour des besoins particuliers.

Demandes liées:

Lié à Publik - Development #49206: Édition riche

Fermé

08 décembre 2020

Historique

#1 - 19 mai 2021 11:16 - Thomas Jund (congé, retour le 29/04)

J'ai commencé à tester 2 3 trucs.

execCommand

Cette spec nous vient d'un truc dev par Windows pour IE 5.5 et implémenté ensuite différemment par chaque navigateur. Elle est incomplète, bancal

et finalement officiellement devenue obsolète pour être un jour remplacée par <https://w3c.github.io/input-events/>

<https://medium.com/content-uneditable/contenteditable-the-good-the-bad-and-the-ugly-261a38555e9c>

La liste des commands et quelques différences expliquées entre browser :

<https://developer.mozilla.org/en-US/docs/Web/API/Document/execCommand>

On a pas vraiment le choix, il va falloir faire avec cette spec dans un premier temps. Les navigateurs "récents" sont à priori un peu plus unifiés entre eux. Mais faire pour IE11, je vois pas comment rester compatible.

Ensuite, garder à l'esprit qu'on va bricoler des trucs. Et qu'il faudra réécrire tout ça dans quelques années.

#2 - 19 mai 2021 11:22 - Thomas Jund (congé, retour le 29/04)

Coller depuis LibreOffice

Le test qui fait peur. Et en effet ça na pas loupé, coller du contenu depuis LibreOffice dans un [contentEditable] ramène du markup ingérable. J'ai décidé de supprimer toute mise en forme et de ne coller que la version plain text du Press Papier.

```
// Paste only plain text
this.editor_content.addEventListener("paste", function(e) {
  e.preventDefault();
  // get text representation of clipboard
  var text = (e.originalEvent || e).clipboardData.getData('text');
  // insert text manually
  document.execCommand("insertText", false, text);
});
```

#3 - 19 mai 2021 15:50 - Thomas Jund (congé, retour le 29/04)

Différences remarquées entre Chrome et FF :

Lors d'un execCommand "bold", "italic", "underline" sur une selection "text", FF va généré `<u><i>text</i></u>` : il 'wrap' avec la nouvelle commande. chrome va généré l'inverse `<i><u>text</u></i>`

Si le cursor est dans un paragraphe (block créé par défaut avec un retour chariot) et que l'invoque execCommand "insertUnorderedList" pour créer une liste à puce, firefox va remplacer le `<p>` par ``, alors que Chrome va l'insérer à l'intérieur du `<p>` (pas cool du tout)

Toujours les listes

Lors d'une suppression d'une liste, chrome remplace les items de liste par des ` ...
` (merci pour le style inline). FF laisse un texte sans tag avec un `
` à la fin.

#4 - 18 août 2021 17:59 - Thomas Jund (congé, retour le 29/04)

ProseMirror me paraît un bon choix de framework.

<https://prosemirror.net>

#5 - 18 août 2021 19:29 - Frédéric Péters

Il y a derrière une demande pour un regard technique, voir si ça peut être packagé et exploité ?

<https://packages.debian.org/search?keywords=prosemirror> liste une série de paquets, mais ça ne sont pas les dernières versions, et il n'y a pas un module comme prosemirror-view qui apparait essentiel, mais peut-être que ça n'était pas le cas à l'époque, mais peut-être que ça n'est pas packagé parce que c'est une galère et pareil pour les nouvelles versions.

...

Sinon je ne sais pas ce que je peux dire d'autre que dire que sûr ça peut être essayé; il y a une vidéo dans la description du ticket avec quelque chose qui peut peut-être servir de modèle pour une démonstration d'utilisation effective, ou quelque chose d'autre. (par exemple déjà dégager la barre d'outils qui fait vieux cul).

#6 - 19 août 2021 10:28 - Thomas Jund (congé, retour le 29/04)

Il y a derrière une demande pour un regard technique, voir si ça peut être packagé et exploité ?

J'ai posé ça là comme premier choix.

Je me suis focalisé à comparer ce qui existe par rapport à notre besoin.

J'ai trouvé que cette solution était

1. maintenu
2. utilisé sur des gros projets open source
3. bien documenté

<https://www.tiptap.dev/> est basé sur ce framework.

Mais je n'ai pas été plus loin (pas encore testé), d'autres avis et regards techniques comme le packaging sont bienvenues, surtout si c'est un aspect éliminatoire.

#7 - 02 novembre 2021 10:33 - Thomas Jund (congé, retour le 29/04)

Premiers tests de proseMirror cette semaine.

L'API est assez mastoc et pas très simple à prendre en main.

Mais l'outil me semble bien construit.

Toute l'UI est à faire.

Et il est contraignant dans le sens où tout block autorisé au sein de l'éditeur doit être déclaré ainsi que ses enfants. Tout autre markup non déclaré est supprimé au parsing. Ce qui me semble très bien mais cela va compliquer la migration du code accepté de CKEditor vers notre nouvel éditeur.

Je prévois un poc pour le camp.

Le code sera distribué sous forme de modules ES. Je vais pousser le code sur misc-tjund pour le moment.

#8 - 23 novembre 2021 16:54 - Thomas Jund (congé, retour le 29/04)

avant de cleaner le code, faut choisir un petit nom.

Toutes les idées sont bienvenues. Premières propositions :

- Phylly (nom donné par Fred à son poc)
- Fraso (nom donné à mon POC)

#9 - 23 novembre 2021 16:58 - Pierre Cros

Godo

#10 - 26 novembre 2021 12:30 - Thomas Jund (congé, retour le 29/04)

- Statut changé de Nouveau à Information nécessaire

- Assigné à changé de Thomas Jund (congé, retour le 29/04) à Benjamin Dauvergne

1ère version de Godo disponible.

J'ai

- cleané le code du poc présenté au camp
- script pour packager une version distribuée du code JS (concaténation des modules JS sous forme de big fichier) avec npm/rollup
- proposé une première UI.

Code ici <https://git.entrouvert.org/misc-tjund.git/tree/godo/>

Test ici <https://perso.entrouvert.org/~tjund/godo/>

J'assigne à Benj pour un premier regard sur le packaging

#11 - 26 novembre 2021 13:31 - Frédéric Péters

Test ici <https://perso.entrouvert.org/~tjund/godo/>

404.

#12 - 26 novembre 2021 13:53 - Thomas Jund (congé, retour le 29/04)

404

retry

#13 - 26 novembre 2021 14:03 - Pierre Cros

- Fichier bulles-en-double.png ajouté

C'est chouette, peut-être tout à fait prématuré (et si c'est le cas, ne pas perdre de temps à me répondre) mais je joins une capture d'écran avec FX qui montre un "G" et "I" parasites, sans que je sache trop comment j'ai fait pour les obtenir.

#14 - 26 novembre 2021 14:21 - Frédéric Péters

(perso pareil peut-être trop tôt pour ces retours, ou il faudra définir une méthode, tu peux juste dire que tu continues à travailler le style).

- le gras/italique translucide ça me perturbe vraiment, ça fait un truc moche qui se déplace et sur lequel un clic ne donne rien, assez étrange. (oui je comprends l'intention "indicateur", mais je trouve que ça ne marche pas).
- en mode basique à autoriser uniquement des paragraphes la "barre d'outils" pourrait totalement ne pas être là.
- je comprends que le textarea est conservé visible pour debug ça sera bien sûr à éliminer.

Sur les dépendances,

"prosemirror-history": "^1.2.0",

pas dispo

"prosemirror-keymap": "^1.1.4",

pas dispo

"prosemirror-model": "^1.15.0",

dispo dans cette version mais bullseye-backports.

"prosemirror-schema-basic": "^1.1.2",

dispo dans cette version mais bullseye.

"prosemirror-state": "^1.3.4",

dispo dans cette version mais bullseye-backports.

"prosemirror-view": "^1.20.1"

pas dispo.

+ "devDependencies",

"@rollup/plugin-buble": "^0.21.3",

ok dispo dans cette version, bullseye ou buster-backports

"@rollup/plugin-node-resolve": "^13.0.6",

pas dispo dans cette version, 11.0.1 dans buster-backports

"rollup": "^2.26.3"

dispo en 2.38.4 dans buster.

Pour le packaging, à titre d'exemple, il pourrait y avoir <https://sources.debian.org/src/node-prosemirror-markdown/1.6.0-1/debian/>

#15 - 27 novembre 2021 09:10 - Paul Marillonnet

Frédéric Péters a écrit :

- le gras/italique translucide ça me perturbe vraiment, ça fait un truc moche qui se déplace et sur lequel un clic ne donne rien, assez étrange.

Pareil, testé, c'est chouette mais j'ai pas tout de suite compris que c'était des actions qui étaient utilisables uniquement lorsque du texte est sélectionné. (Aussi, sans doute c'est parce que je suis conditionné par les traitements de texte bureautique où ces actions sont utilisables même lorsque du texte n'est pas sélectionné, et dans ce cas là c'est le texte qui va être écrit à cet endroit précis qui prend cette mise en forme. Peut-être que pour godo il y a moyen de retirer complètement ces deux icônes gras/italique lorsqu'aucun texte n'est sélectionné ?)

#16 - 29 novembre 2021 10:13 - Thomas Jund (congé, retour le 29/04)

qui montre un "G" et "I" parasites

C'est bien un bug qu'il faudra régler dans une configuration multi-editeur dans une même page (si on garde ce menu fantôme)

le gras/italique translucide ça me perturbe vraiment
en mode basique à autoriser uniquement des paragraphes la "barre d'outils" pourrait totalement ne pas être là.

J'y ai pensé, à il y a alors cet aspect UI à régler :

Si pas de nav "blocks" ni nav "inline" apparente par défaut (parce que uniquement visible lors d'une sélection de texte), quel indicateur donné aux usagers qu'ils ont la possibilité de formater leur texte ?

les traitements de texte bureautique où ces actions sont utilisables même lorsque du texte n'est pas sélectionné

Ils fonctionnent sans sélection, mais la vue n'est pas mise à jour (création de balises strong ou em) tant qu'un caractère n'est pas saisi, je n'ai donc pas réussi à retourner un feedback pour informer que l'action est prise en compte.

#17 - 10 décembre 2021 19:29 - Benjamin Dauvergne

Packaging released, mail envoyé sur trav@ :

J'ai fini le packaging godo via deux dépôts :
* <https://git.entrouvert.org/godo.js.git> qui contient le code source de base et produit un paquet libjs-godo
* <https://git.entrouvert.org/debian/xstatic-godo.git> qui intègre tout ça à la machinerie XStatic qu'on utilise habituellement.

Le paquet godo.js.git est intégré à jenkins classiquement, le paquet xstatic-godo par contre il faut l'exécuter via le jobs "debs" mais il ne nécessite pas d'être mis à jour à chaque fois car il ne contient rien qui dépende du dépôt godo.

Au niveau utilisation il suffit de s'inspirer de l'exemple donné par tjund[1] et remplacer les URLs par "% xstatic 'godo' 'godo{[.min].js,.css}' %".

Paquets disponibles en buster et bullseye, eobuilder et testing[2]

[1]: <https://git.entrouvert.org/godo.js.git/tree/dist/index.html>
[2]: <https://deb.entrouvert.org/packages.html#libjs-godo>
<https://deb.entrouvert.org/packages.html#python3-xstatic-godo>

#18 - 11 décembre 2021 11:05 - Frédéric Péters

- Lié à *Development #49206: Édition riche ajouté*

#19 - 11 décembre 2021 11:06 - Frédéric Péters

- *Statut changé de Information nécessaire à Fermé*

Suivi désormais dans <https://dev.entrouvert.org/projects/godo/issues/> je vais y faire des tickets pour certaines des remarques.

Fichiers

bulles-en-double.png

17,5 ko 26 novembre 2021

Pierre Cros