1.15

TEMPS 1 • LE PORTFOLIO • SÉANCE 4

L'éditeur de texte riche

Thibault Six • IFOSUP Wavre • 2026-2027



? Un éditeur va remplir content. Qui peut y écrire, et qu'y glisserait un visiteur malveillant ?

Seul l'auteur connecté, vous, écrit dans content ; un visiteur qui écrirait du HTML pourrait y glisser du JavaScript.

resources/js/pages/projects/Show.vue

```
<div v-html="project.content"></div>
```

écrite au chapitre 1.07 : elle affiche le HTML de la colonne `content` tel qu'il est en base



La même ligne affiche ce qu'on lui donne. En haut, c'est vous qui écrivez ; en bas, ce serait n'importe qui.

1.15

? Depuis le back-office, pourquoi un textarea ne suffit-il pas pour écrire le contenu d'un projet ?

Un textarea ne produit que du texte brut ; un éditeur de texte riche écrit lui-même les balises, comme .

au chapitre 1.06 : du HTML écrit dans une chaîne PHP, une apostrophe échappée dans chaque d'interfaces

UN TEXTAREA

```
<h2>Le projet</h2>
<p>Un texte <strong>important</strong>.</p>
```

du texte brut : les balises se tapent à la main

la moindre balise oubliée casse la mise en page de la fiche

UN ÉDITEUR DE TEXTE RICHE



il met en forme pendant que vous écrivez, comme un traitement de texte

l'éditeur écrit lui-même la balise

Tiptap : une bibliothèque d'éditeur pour Vue.
Le composant est fourni : vous l'installez, vous le branchez, vous comprenez ce qu'il envoie.

À gauche, c'est vous qui tapez les balises ; à droite, c'est l'éditeur. Dans les deux cas, content reçoit du HTML.

? Quels paquets faut-il installer pour l'éditeur, et pour mettre en forme la fiche ?

Trois paquets Tiptap font l'éditeur, un quatrième met en forme la fiche ; ne lancez pas npm audit fix --force.



Suivez la ligne : arrêter, installer, relancer. Le message de npm, en bas, ne demande aucune action.

04

Le composant fourni fait plus de 150 lignes. Lesquelles faut-il lire pour s'en servir ?

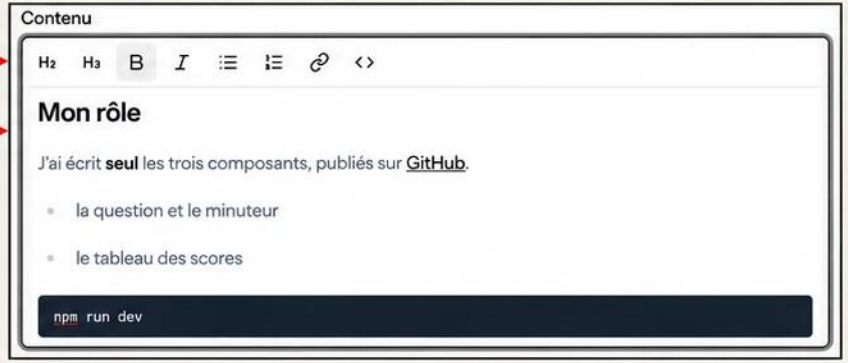
Le composant est fourni : lisez ses props, useEditor(), onUpdate et la barre de huit boutons, sans tout comprendre.

resources/js/components/RichTextEditor.vue

① **defineProps**	**name**	le nom du champ envoyé au serveur
	defaultValue	le HTML de départ
	label	le nom que lit un lecteur d'écran
② **useEditor()**	crée l'éditeur ; `StarterKit.configure()` garde les titres de niveaux 2 et 3, et désactive le barré, le souligné et la ligne horizontale	
③ **onUpdate**	s'exécute à chaque frappe et recopie le HTML dans la variable `html`	
④ **buttons**	la barre : huit boutons, rendus par un `v-for`, chacun avec `type="button"`	

vous le créez en copiant le fichier en entier : vous ne l'écrivez pas

le composant dans le formulaire de modification



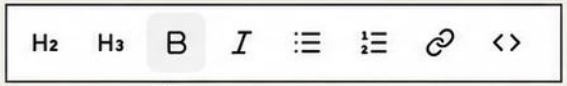
Quatre passages suffisent pour se servir du composant. Les numéros suivent l'ordre du fichier, de haut en bas.

1.15

Vous cliquez sur « Gras ». Qui écrit la balise dans le contenu ?

Chaque bouton de la barre appelle une commande de Tiptap, et Tiptap écrit la balise : toggleBold() pose .

la barre, « Gras » allumé



un clic

resources/js/components/RichTextEditor.vue — extrait

```
{
    label: 'Gras',
    icon: Bold,
    active: () => editor.value.isActive('bold'),
    run: () => editor.value.chain().focus().toggleBold().run(),
},
```

Tiptap écrit la balise ****

active : le bouton reste allumé quand le curseur est dans du gras

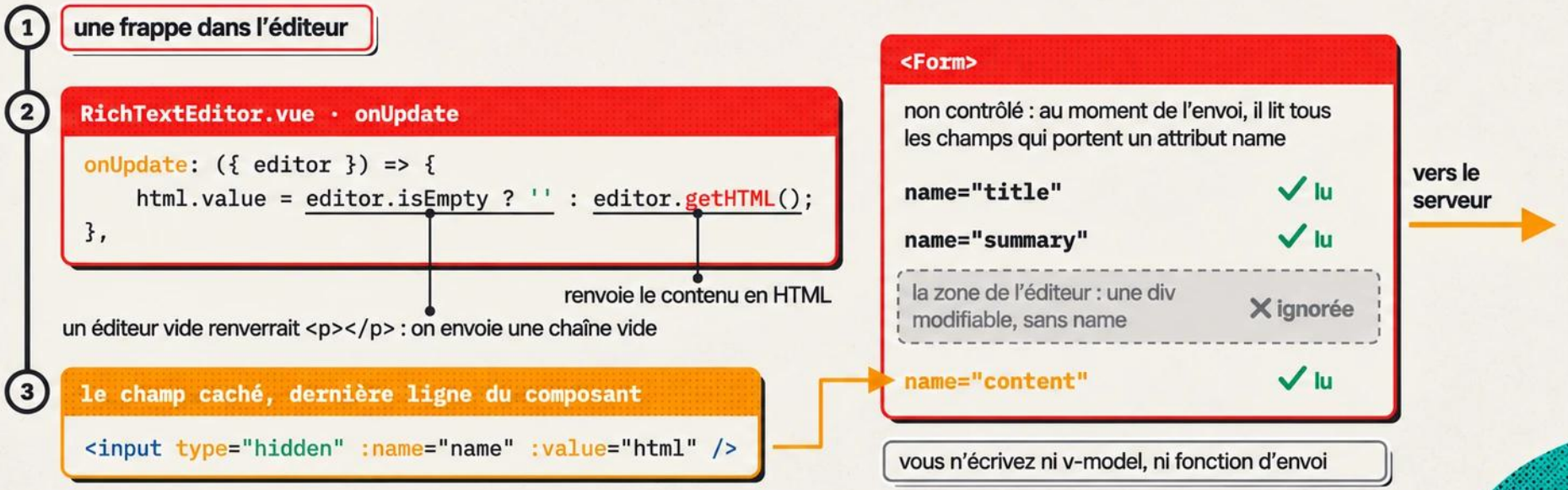
LES HUIT BOUTONS ET LEURS BALISES

- Titre 2 → <h2>
- Titre 3 → <h3>
- Gras →
- Italique →
- Liste à puces, Liste numérotée → des listes
- Lien → <a>
- Bloc de code → <pre><code>

Suivez le clic : le bouton appelle sa fonction, Tiptap écrit la balise. Les sept autres boutons suivent le même chemin.

? La zone de l'éditeur n'est pas un champ de formulaire. Comment son HTML arrive-t-il au serveur ?

<Form> ne lit que les champs qui ont un name : le composant recopie son HTML dans un champ caché nommé content.



Suivez la frappe : onUpdate recopie le HTML, le champ caché le porte, <Form> le lit avec les autres champs.

Que faut-il ajouter dans Create.vue et dans Edit.vue pour brancher l'éditeur ?

Les deux pages reçoivent le composant ; Edit.vue lui passe en plus le contenu actuel du projet par :default-value.

```
import RichTextEditor from '@/components/RichTextEditor.vue';
```

dans les deux pages, avec les autres imports

resources/js/pages/admin/projects/Create.vue

```
<div class="grid gap-2">
    <Label>Contenu</Label>
    <RichTextEditor name="content" label="Contenu du projet" />
    <InputError :message="errors.content" />
</div>
```

entre le résumé et le lien de la démo

pas de `for` : une div modifiable ne peut pas être visée par une étiquette ; `role="textbox"` et `aria-label` la font annoncer

extrait

resources/js/pages/admin/projects/Edit.vue

```
<RichTextEditor
    name="content"
    label="Contenu du projet"
    :default-value="project.content"
/>
```

l'éditeur s'ouvre avec le contenu actuel du projet

content absent de l'only() de edit() : l'éditeur s'ouvre vide, et enregistrer efface le contenu

côté serveur, la règle existe depuis le chapitre 1.12 : `'content' => ['nullable', 'string']`

Les deux pages reçoivent le même bloc. Seule la page de modification donne au composant un HTML de départ.

? Vous envoyez le formulaire avec un titre vide. Que contient le corps de la requête ?

L'éditeur envoie du HTML dans la clé content, à côté des autres champs, et le serveur l'enregistre tel quel.

1. « Titre 2 », puis « Le projet »
2. un mot en gras : Ctrl+B (Cmd+B sur Mac)
3. titre et résumé vides, envoyer

LE CORPS DE LA REQUÊTE, RELEVÉ DANS LE NAVIGATEUR

```
POST http://…/admin/projets
{"title":"","summary":"","content":"<h2>Le projet</h2><p>Un texte <strong>important</strong>.</p>…","demo_url":"","repo_url":""}
```

le titre · le paragraphe · le gras

le formulaire, après l'envoi

Le champ titre est obligatoire.
Le champ résumé est obligatoire.

le texte de l'éditeur reste en place : la page n'a pas été rechargée

le serveur : rien ne change

`'content' => ['nullable', 'string']` → `$project->update()` → **colonne content**

le HTML est enregistré tel qu'il est arrivé ; un éditeur vidé envoie une chaîne vide, que Laravel transforme en null

Lisez le ticket : content est une clé comme les autres, et sa valeur est du HTML. Ses balises sont celles de la barre.

? Sur la fiche publique, le titre a la taille du texte et la liste n'a pas de puces. Pourquoi ?

Tailwind efface les styles du navigateur ; la classe prose, fournie par le plugin typography, les redonne au HTML.

resources/css/app.css

```
@plugin '@tailwindcss/typography';
```

Tailwind 4 : un plugin se déclare dans le CSS, juste après les `@import` ; pas de `tailwind.config.js`

resources/js/pages/projects/Show.vue

```
<div class="prose dark:prose-invert" v-html="project.content"></div>
```

`dark:prose-invert` inverse les couleurs quand le kit pose la classe `dark`

SANS PROSE

Mon rôle

J'ai écrit **seul** les trois composants, publiés sur GitHub.

la question et le minuteur

le tableau des scores

`npm run dev`

le titre a la taille du texte, la liste n'a pas de puces, le code ressemble à un paragraphe

prose

AVEC PROSE

Mon rôle

J'ai écrit **seul** les trois composants, publiés sur GitHub.

- la question et le minuteur
- le tableau des scores

```
npm run dev
```

tailles de titres, marges, puces, liens soulignés, fond pour le code

Même contenu, même page : seule la classe prose change. Comparez la taille du titre et les puces.

1.15

? v-html affiche le HTML de content sans filtre. Pourquoi est-ce acceptable sur votre portfolio ?

v-html convient ici parce que vous seul, connecté, écrivez dans content ; le HTML d'un visiteur se nettoie côté serveur.

`v-html` **insère le HTML de la colonne tel quel, sans filtre**

ACCEPTABLE ICI

- au chapitre 1.07 : le HTML de votre seeder
- aujourd'hui : le HTML de l'éditeur
- `v-html`
- la fiche

- ☑ **un seul auteur écrit dans `content`**
- ☑ **il doit être connecté**
- ☑ **c'est vous**

À NETTOYER

un visiteur peut écrire du HTML montré à d'autres

un commentaire | un message | un profil

↓

nettoyé côté serveur avant d'être enregistré → `v-html`

XSS **sinon, n'importe qui fait exécuter son JavaScript dans le navigateur de vos lecteurs**

À gauche, trois conditions toutes vraies. Dès qu'une seule tombe, la colonne de droite s'applique.

Le texte tapé disparaît, « Gras » enregistre le projet, la fiche n'a aucun style : pourquoi ?

Wayfinder recharge la page et efface le texte, un bouton sans type envoie le formulaire, et sans @plugin, prose ne fait rien.

Le texte tapé disparaît pendant un essai

```
Types generated for actions, routes, form variants
```

enregistrer un fichier PHP de `app/Http` fait régénérer Wayfinder, et le navigateur recharge la page

modifiez vos règles de validation avant de remplir un long contenu, pas pendant

Un clic dans la barre envoie le formulaire

RichTextEditor.vue — extrait

```
<Button
    v-for="button in buttons"
    :key="button.label"
    type="button"
    variant="ghost"
    size="icon-sm"
```

le `Button` de shadcn-vue rend un `<button>` sans type, et un bouton sans type, dans un formulaire, envoie le formulaire

recopiez le fichier en entier

La fiche n'a aucun style

```
@plugin '@tailwindcss/typography';
```

le plugin n'est pas déclaré, ou `composer run dev` n'a pas été relancé après `npm install`

vérifiez la ligne `@plugin` de `app.css`, puis relancez

Lisez chaque colonne de haut en bas : ce que vous voyez, ce qui le cause, ce que vous faites.

? Qu'est-ce qui doit être vrai à la fin ?

À vous, vingt-cinq minutes : le contenu de vos trois projets, réécrit en texte riche depuis le back-office.

À FAIRE

1. Les trois paquets de Tiptap et `@tailwindcss/typography`, sans `npm audit fix --force`.
2. `RichTextEditor.vue` copié en entier.
3. Le bloc « Contenu » dans `Create.vue`, et dans `Edit.vue` avec `:default-value="project.content"`.
4. La ligne `@plugin` dans `app.css`, et les classes `prose dark:prose-invert` sur la fiche.
5. Le contenu de vos trois projets réécrit depuis le back-office, avec un titre de niveau 2 par rubrique de votre fiche projet : contexte, rôle, problème, choix, résultat, ce que je referais.

À VÉRIFIER

25 min

- [ ] La barre a huit boutons, et aucun n'envoie le formulaire.
- [ ] La modification d'un projet ouvre l'éditeur avec le contenu existant.
- [ ] Un envoi avec un titre vide garde le texte de l'éditeur.
- [ ] La fiche publique montre des titres, des listes à puces et un lien, mis en forme par `prose`.

Cinq étapes à gauche, quatre vérifications à droite ; la troisième vérification se fait avec un titre laissé vide.

1.15