

ac_text_to_html

Classe JS autonome — Éditeur WYSIWYG pour non-techniciens

Jean-Noël Vidallet · artisan-code.fr · Mars 2026

1. Présentation

ac_text_to_html est une classe JavaScript autonome qui transforme un textarea HTML en éditeur de texte mis en forme (WYSIWYG). Elle est conçue pour un public de non-techniciens qui veulent mettre en page du texte sans connaître le HTML.

Le résultat est stocké dans un <input hidden> sous forme de HTML propre, prêt à être envoyé au serveur.

Cas d'usage

- Rédaction d'un article de blog par un auto-entrepreneur
- Saisie d'un commentaire par un visiteur
- Tout champ textarea dans une grille ac_grille_data_table (via le paramètre widget)

Caractéristiques principales

- Classe JS unique — CSS et Google Fonts embarqués, zéro dépendance
- Remplace visuellement le textarea — l'utilisateur ne voit jamais de balises HTML
- Stocke le HTML généré dans un <input hidden> synchronisé en temps réel
- Charge les valeurs initiales en HTML (mode édition d'un enregistrement existant)
- Compteur de caractères obligatoire avec alerte visuelle et blocage à la limite
- Polices web-safe système + 17 Google Fonts groupées par catégorie
- 70 smileys en 6 catégories insérables au curseur
- Upload d'images avec redimensionnement automatique (délégué à ac_fichier_upload)
- Copyright discret cliquable vers artisan-code.fr

2. Installation et fichiers

Fichier	Destination
ac_text_to_html.js	/assets/progiciel/classes/js/ac_text_to_html.js
ac_fichier_upload.js	/assets/progiciel/classes/js/ac_fichier_upload.js (requis pour les images)

Inclusion dans la page — ac_fichier_upload.js DOIT être chargé en premier :

```
<script src="/assets/progiciel/classes/js/ac_fichier_upload.js"></script>
<script src="/assets/progiciel/classes/js/ac_text_to_html.js"></script>
```

⚠ Si upload_url n'est pas utilisé, ac_fichier_upload.js est facultatif. Le bouton image n'apparaît pas.

3. Utilisation minimale

Dans le HTML, placer un textarea avec un id unique :

```
<textarea id="mon_texte" name="mon_texte"></textarea>
```

Instancier la classe après le chargement du DOM :

```
const editeur = new ac_text_to_html('mon_texte');
```

La classe crée automatiquement un `<input hidden>` idé "mon_texte_html" qui contiendra le HTML généré. C'est cet input qui est envoyé au serveur lors du submit du formulaire.

4. Options de configuration

Toutes les options sont facultatives — les valeurs par défaut sont prêtes à l'emploi.

Option	Type	Défaut	Description
largeur	string	"100%"	Largeur CSS du composant
hauteur	string	"280px"	Hauteur minimale de la zone de saisie
maxlength	int	5000	Nombre maximum de caractères (texte brut, sans balises). Active le compteur.
upload_url	string	null	URL du script PHP de réception des images. Active le bouton  . null = bouton absent.
img_largeur	int	600	Largeur par défaut des images insérées : 600, 400 ou 200 (px). L'utilisateur peut changer dans le panneau.
callback	function	null	Fonction appelée à chaque modification : fn(html). Reçoit le HTML généré.

Exemple avec toutes les options

```
const editeur = new ac_text_to_html('mon_texte', {
  largeur      : '100%',
  hauteur      : '320px',
  maxlength    : 5000,
  upload_url   : 'upload_image.php',
  img_largeur  : 600,
  callback     : function(html) {
    document.getElementById('apercu').innerHTML = html;
  }
});
```

5. Barre d'outils

Groupe	Outil	Description
Format	Gras	Ctrl+B — met le texte sélectionné en gras
Format	Italique	Ctrl+I — met le texte sélectionné en italique
Format	Souligné	Ctrl+U — souligne le texte sélectionné
Police	Police	Sélection parmi 13 polices système + 17 Google Fonts groupées
Police	Taille	De "Très petite" à "Maximale" (7 niveaux)
Couleur	Couleur texte	Couleur du texte sélectionné — color picker natif (toutes couleurs)
Couleur	Couleur fond	Couleur de surlignage — color picker natif (toutes couleurs)
Alignement	Gauche	Aligne le paragraphe à gauche
Alignement	Centré	Centre le paragraphe
Alignement	Droite	Aligne le paragraphe à droite
Divers	Lien	Crée un lien cliquable (s'ouvre dans un nouvel onglet)
Divers	Bloc citation	Entoure la sélection d'un bloc citation mis en forme
Divers	Image 	Ouvre le panneau d'insertion d'image. Visible si upload_url est défini. Grisé avec croix rouge × si ac_fichier_upload.js est absent.
Divers	HTML </>	Ouvre l'éditeur HTML dual-screen (rendu 2026 + terminal années

Groupe	Outil	Description
		80). Sync temps réel.
Divers	Effacer format	Supprime toute mise en forme du texte sélectionné. Toujours en dernière position.

6. Polices disponibles

Polices système (web-safe)

Arial, Arial Black, Comic Sans MS, Courier New, Georgia, Impact, Lucida Console, Lucida Sans, Palatino, Tahoma, Times New Roman, Trebuchet MS, Verdana

Google Fonts — Sans-serif

Roboto, Open Sans, Lato, Montserrat, Nunito, Raleway, Oswald

Google Fonts — Serif

Playfair Display, Merriweather, Lora, EB Garamond

Google Fonts — Manuscriptes

Dancing Script, Pacifico, Caveat, Satisfy

Google Fonts — Monospace

Roboto Mono, Source Code Pro

 Les Google Fonts sont chargées automatiquement par la classe via CDN. La page hôte n'a rien à faire.

7. Compteur de caractères

Le compteur est toujours affiché dans le footer du composant. Il porte sur le texte brut (sans les balises HTML), ce qui est plus parlant pour l'utilisateur.

État	Condition	Comportement
Normal	Moins de 90% de la limite	Compteur gris discret
Alerte	Entre 90% et 100%	Compteur orange, texte en gras
Limite	100% atteint	Compteur rouge, saisie bloquée

Valeurs recommandées pour AE-Pocket

Contexte	maxlength recommandé
Article de blog	50 000
Commentaire visiteur	2 000
Champ textarea dans une grille	1 000

8. Méthodes publiques

Méthode	Retour	Description
getHTML()	string	Retourne le HTML généré (contenu de l'input hidden)
setHTML(html)	void	Injecte du HTML dans l'éditeur — utile en mode édition d'un enregistrement existant
clear()	void	Vide complètement l'éditeur et l'input hidden
getTexteCourtTronque(n=100)	string	Retourne le texte brut tronqué à n caractères, suivi de ... si tronqué. Utilisé par ac_grille_data_table pour l'affichage en cellule.

Exemple d'utilisation des méthodes

```
// Charger un enregistrement existant
editeur.setHTML(data.description);

// Lire le HTML avant envoi au serveur
const html = editeur.getHTML();

// Affichage tronqué dans une grille (appelé par ac_grille_data_table)
const resume = editeur.getTexteCourtTronque(100);
```

9. Insertion d'images

Quand upload_url est défini, un bouton  apparaît dans la barre d'outils. Il ouvre un panneau qui permet de choisir la largeur puis d'uploader une image via ac_fichier_upload.

Fonctionnement

- Clic sur  → panneau avec sélecteur de largeur (Grande 600px / Moyenne 400px / Petite 200px)
- Drop ou sélection d'une image → upload AJAX immédiat vers upload_url
- Upload réussi → image insérée au curseur avec max-width:100% + largeur choisie
- Panneau fermé automatiquement après insertion
- Les outils d'alignement fonctionnent sur l'image (centre le paragraphe contenant l'image)

HTML généré

```

```

Script PHP de réception (upload_url)

```
<?php
header('Content-Type: application/json');
$fichier = $_FILES['fichier'];
$ext      = strtolower(pathinfo($fichier['name'], PATHINFO_EXTENSION));
$autorise = ['jpg', 'jpeg', 'png', 'gif', 'webp'];
if (!in_array($ext, $autorise)) {
    echo json_encode(['ok'=>false, 'message'=>'Type non autorisé']);
    exit;
}
$nom = uniqid('img_') . '.' . $ext;
move_uploaded_file($fichier['tmp_name'], 'uploads/' . $nom);
echo json_encode(['ok'=>true, 'chemin'=>'uploads/' . $nom]);
```

⚠ Si `ac_fichier_upload.js` est absent, le bouton image  apparaît quand même mais grisé avec une croix rouge ✕ en badge. Il est désactivé et affiche un tooltip explicatif au survol.

10. Intégration avec `ac_grille_data_table`

La classe s'intègre dans `ac_grille_data_table` via le paramètre widget d'une colonne de type textarea. Ce paramètre est prévu mais pas encore implémenté dans `ac_grille_data_table`.

Configuration prévue (non encore implémentée)

```
[
  'titre'  => 'Description',
  'champ'  => 'description',
  'type'    => 'textarea',
  'widget' => 'ac_text_to_html', // nom exact de la classe JS
]
```

En mode affichage dans la grille, `ac_grille_data_table` appellera `getTexteCourtTronque(100)` pour afficher les 100 premiers caractères du texte brut dans la cellule.

⚠ *widget est une propriété prévue pour `ac_grille_data_table` — elle n'est pas encore implémentée dans cette classe.*

11. L'input hidden

La classe crée automatiquement un `<input hidden>` dans le DOM, inséré après le textarea original. Son id et son name sont construits à partir de l'id du textarea.

Attribut	Valeur
type	hidden
id	textareaid + "_html" (ex: "mon_texte_html")
name	name du textarea original
value	HTML généré, synchronisé en temps réel

⚠ *C'est l'input hidden qui doit être lu côté serveur (PHP), pas le textarea original qui reste vide.*

Lecture côté PHP

```
$description = $_POST['mon_texte_html']; // contient le HTML propre
```

⚠ *Toujours assainir le HTML reçu côté serveur avant stockage en base (HTMLPurifier ou strip_tags avec liste blanche).*