# ac_timeline

**Classe JS autonome — Frise chronologique**

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

---

## *1. Présentation*

ac_timeline est une classe JavaScript autonome qui génère une frise chronologique interactive — verticale ou horizontale — à partir d'un tableau d'événements. Elle gère le tri par date, les animations au scroll, les types colorés avec icônes, le callback au clic et l'ajout dynamique d'événements.

> ⚠️ Cette classe est en version 1.0 — le rendu visuel et les options sont susceptibles d'évoluer selon les besoins des projets.

Caractéristiques principales :

- Classe JS unique — CSS entièrement embarqué
- Zéro dépendance externe
- Orientation verticale (alternance gauche/droite) ou horizontale (scroll)
- Tri par date ascendant ou descendant
- Animation au scroll via IntersectionObserver
- Types prédéfinis avec couleur et icône (contact, devis, paiement, incident, rdv, document, note)
- Couleur personnalisable par événement
- Callback au clic sur un événement
- Ajout dynamique via ajouterEvenement()
- Convention Classothèque — setEvenements() / getEvenements()

Cas d'usage :

- Historique client MonEspace-Pro (paiements, contacts, devis, incidents)
- Parcours bénéficiaire DEFI (arrivée, ateliers, bilans)
- Jalons d'un projet (horizontal)
- Timeline de facturation

---

## *2. Installation et fichiers*

| Fichier | Emplacement |
|---|---|
| ac_timeline.js | /assets/classes/js/ac_timeline.js |

Inclusion dans la page (avant `</body>`) :

```
<script src="/assets/classes/js/ac_timeline.js"></script>
```

Dans le HTML, placer un div conteneur :

```
<div id="ma_timeline"></div>
```

---

## 3. Utilisation minimale

```
new ac_timeline('ma_timeline', {
    tri: 'desc'
}).setEvenements([
    { date: '2026-03-15', type: 'paiement', titre: 'Paiement reçu',
description: '12€/mois' },
    { date: '2026-02-01', type: 'contact',  titre: 'Premier contact',
description: 'Via formulaire' },
]);
```

---

## 4. Options de configuration

| Option | Type | Défaut | Description |
|---|---|---|---|
| orientation | string | 'vertical' | 'vertical' ou 'horizontal' |
| tri | string | 'desc' | 'asc' (chronologique) ou 'desc' (plus récent en premier) |
| animation | bool | true | Animation d'apparition au scroll (IntersectionObserver) |
| callback | function | null | fn(evenement) — appelée au clic sur une carte |
| couleur_ligne | string | '#e5e7eb' | Couleur de la ligne centrale |
| largeur | string | '100%' | Largeur CSS du composant |
| couleurs_types | object | voir ci-dessous | Mapping type → couleur pastille |
| icones_types | object | voir ci-dessous | Mapping type → icône emoji |

### Types prédéfinis

| Type | Couleur | Icône |
|---|---|---|
| contact | #3b82f6 (bleu) | 📞 |
| devis | #f59e0b (orange) | 📄 |
| paiement | #22c55e (vert) | 💶 |
| incident | #ef4444 (rouge) | ⚠️ |

| Type | Couleur | Icône |
|---|---|---|
| rdv | #8b5cf6 (violet) | 📅 |
| document | #06b6d4 (cyan) | 📎 |
| note | #e8620a (orange artisan) | ✏️ |
| default | #6b7280 (gris) | ● |

Les couleurs_types et icones_types sont fusionnés avec les défauts — il suffit de surcharger les types souhaités.

---

## *5. Structure d'un événement*

| Propriété | Type | Requis | Description |
|---|---|---|---|
| date | string | oui | Date ISO (YYYY-MM-DD) ou toute chaîne parseable par new Date() |
| titre | string | oui | Titre principal de la carte |
| description | string | non | Texte secondaire affiché sous le titre |
| type | string | non | Type prédéfini (contact, devis, paiement…) — détermine couleur et icône |
| couleur | string | non | Couleur CSS directe — surcharge la couleur du type |

---

## *6. Méthodes publiques*

| Méthode | Retour | Description |
|---|---|---|
| setEvenements(tableau) | void | Charge tous les événements et redessine |
| ajouterEvenement(evt) | void | Ajoute un événement et redessine |
| getEvenements() | array | Retourne le tableau des événements courants |
| vider() | void | Vide la timeline |

---

## 7. Orientation verticale

La frise verticale alterne les événements gauche/droite autour d'une ligne centrale. C'est le mode le plus adapté pour les historiques longs.

```
new ac_timeline('t1', {
    orientation : 'vertical',
    tri         : 'desc',
    callback    : function(evt) {
        console.log('Cliqué :', evt.titre);
    }
}).setEvenements(historique);
```

---

## 8. Orientation horizontale

La frise horizontale place les événements au-dessus et en dessous d'une ligne horizontale, avec scroll automatique si le contenu dépasse. Idéale pour les jalons de projet.

```
new ac_timeline('t2', {
    orientation : 'horizontal',
    tri         : 'asc',
}).setEvenements(jalons);
```

---

## 9. Callback au clic

Quand `callback` est défini, chaque carte devient cliquable (curseur pointer, hover animé). La fonction reçoit l'objet événement complet.

```
new ac_timeline('t1', {
    callback: function(evt) {
        // evt = { date, titre, description, type, couleur }
        document.getElementById('detail').innerHTML =
            '<strong>' + evt.titre + '</strong><br>' + evt.description;
    }
});
```

---

## 10. Ajout dynamique

```
const tl = new ac_timeline('t1');
tl.setEvenements(evenementsInitiaux);

// Après une action utilisateur
tl.ajouterEvenement({
    date        : new Date().toISOString().slice(0, 10),
    type        : 'paiement',
    titre       : 'Nouveau paiement',
    description : '12€ — abonnement Pro'
});
```

## 11. Personnalisation des types

Surcharger uniquement les types souhaités — les autres gardent leurs valeurs par défaut :

```
new ac_timeline('t1', {
    couleurs_types: {
        'paiement' : '#10b981',   // vert plus vif
        'incident' : '#dc2626',   // rouge plus sombre
        'facture'  : '#7c3aed',   // type personnalisé
    },
    icones_types: {
        'paiement' : '€',
        'facture'  : '🧾',
    }
});
```

## 12. Convention Classothèque

| Méthode | Description |
|---|---|
| setEvenements(tableau) | Charge les données |
| getEvenements() | Retourne les données courantes |

## 13. Bonnes pratiques

- Les dates doivent être au format ISO YYYY-MM-DD pour un tri correct
- En orientation horizontale, prévoir un conteneur avec overflow:hidden pour masquer la scrollbar
- Le callback est optionnel — sans lui les cartes sont en lecture seule
- Pour une timeline très longue, préférer l'orientation verticale avec animation au scroll
- ajouterEvenement() retrie automatiquement selon l'option tri définie