

ac_agenda

Classe JS autonome — Agenda interactif

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

1. Présentation

ac_agenda est une classe JavaScript autonome qui génère un agenda interactif complet avec quatre vues (jour, semaine, mois, année), une modale de création/édition/suppression de rendez-vous, et le drag & drop. La classe ne gère aucun backend — elle dispatche des événements custom que la page appelante branche sur ses propres appels serveur.

Caractéristiques principales :

- Classe JS unique — CSS entièrement embarqué
 - Zéro dépendance externe
 - 4 vues : jour, semaine, mois, année
 - Navigation par boutons < >, bouton Aujourd'hui, clic sur les en-têtes
 - Modale création — clic sur case vide (titre, date, heures, description, couleur)
 - Modale édition — clic sur un RDV existant (mêmes champs + bouton Supprimer)
 - Palette de 10 couleurs personnalisable
 - Drag & drop des RDV avec event acagenda:deplacer
 - RDV multi-jours (date_debut + date_fin)
 - RDV journée entière (sans heure)
 - Zéro couplage backend — 4 événements custom dispatché sur le container
 - API publique complète — setRdvs, ajouterRdv, modifierRdv, supprimerRdv
 - Responsive — vue année 2 colonnes sur mobile
-

2. Installation et fichiers

Fichier	Emplacement
ac_agenda.js	/assets/classes/js/ac_agenda.js

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

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

Dans le HTML, placer un div conteneur :

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

3. Utilisation minimale

```
const agenda = new ac_agenda('mon_agenda');

// Charger les RDV
agenda.setRdvs(rdvs);

// Brancher les événements
document.getElementById('mon_agenda').addEventListener('acagenda:creer', e =>
{
    // fetch POST serveur, puis :
    agenda.ajouterRdv({ id: newId, ...e.detail });
});
```

4. Options de configuration

Option	Type	Défaut	Description
vue	string	'mois'	Vue initiale : 'jour', 'semaine', 'mois', 'annee'
rdvs	array	[]	RDV chargés au démarrage
granularite	number	30	Créneaux en minutes : 15, 30 ou 60
heure_debut	string	'08:00'	Début de la plage horaire (vues jour et semaine)
heure_fin	string	'20:00'	Fin de la plage horaire
couleur	string	'#5a7a62'	Couleur par défaut des RDV
lecture_seule	bool	false	Désactive les clics et le drag & drop
largeur	string	'100%'	Largeur CSS du composant
couleurs_palette	array	voir ci-dessous	10 couleurs proposées dans la modale
callback_charger	function	null	fn(date_debut, date_fin) — appelée à chaque navigation

Palette par défaut :

```
['#5a7a62', '#1a3d6b', '#e8620a', '#e53e3e', '#805ad5', '#d69e2e', '#2b6cb0', '#276749', '#c05621', '#702459']
```

5. Structure d'un RDV

Propriété	Type	Requis	Description
id	any	oui	Identifiant unique (number ou string)
titre	string	oui	Titre du rendez-vous
date_debut	string	oui	Date ISO YYYY-MM-DD
date_fin	string	non	Date ISO YYYY-MM-DD — RDV multi-jours (si absent = date_debut)
heure_debut	string	non	Heure HH:MM — si absent = RDV journée entière
heure_fin	string	non	Heure HH:MM de fin
description	string	non	Description libre
couleur	string	non	Couleur CSS — surcharge la couleur par défaut

 La propriété canonique pour l'heure est `heure_debut`. La classe accepte aussi `heure` pour la compatibilité avec la v1 de l'agenda.

6. Événements custom

La classe dispatche ses événements sur le container DOM. La page appelante branche ces événements pour appeler son backend, puis met à jour l'agenda via l'API publique.

Événement	Détail	Description
acagenda:creer	{ titre, date_debut, heure_debut, heure_fin, description, couleur }	Clic sur Créer dans la modale
acagenda:modifier	{ id, titre, date_debut, heure_debut, heure_fin, description, couleur }	Clic sur Enregistrer dans la modale
acagenda:supprimer	{ id }	Clic sur Supprimer dans la modale
acagenda:deplacer	{ id, ancienne_date, ancienne_heure, nouvelle_date, nouvelle_heure }	Drop après drag & drop

Exemple de branchement complet

```
const container = document.getElementById('mon_agenda');
```

```

container.addEventListener('acagenda:creer', async e => {
  const res = await fetch('/api/rdv', { method:'POST', body:
JSON.stringify(e.detail) });
  const data = await res.json();
  agenda.ajouterRdv({ id: data.id, ...e.detail, date_debut:
e.detail.date_debut });
});

container.addEventListener('acagenda:modifier', async e => {
  await fetch('/api/rdv/' + e.detail.id, { method:'PUT', body:
JSON.stringify(e.detail) });
  agenda.modifierRdv({ ...rdvExistant, ...e.detail });
});

container.addEventListener('acagenda:supprimer', async e => {
  await fetch('/api/rdv/' + e.detail.id, { method:'DELETE' });
  agenda.supprimerRdv(e.detail.id);
});

container.addEventListener('acagenda:deplacer', async e => {
  await fetch('/api/rdv/' + e.detail.id, { method:'PATCH', body:
JSON.stringify({
    date_debut : e.detail.nouvelle_date,
    heure_debut : e.detail.nouvelle_heure,
  })});
  // Mettre à jour localement
  const rdv = agenda._rdvs.find(r => r.id === e.detail.id);
  if (rdv) {
    rdv.date_debut = e.detail.nouvelle_date;
    rdv.heure_debut = e.detail.nouvelle_heure;
    agenda.setRdvs([...agenda._rdvs]);
  }
});

```

7. API publique

Méthode	Description
setRdvs(rdvs)	Remplace tous les RDV et rafraîchit l’affichage
ajouterRdv(rdv)	Ajoute un RDV et rafraîchit (appeler après INSERT serveur)
modifierRdv(rdv)	Met à jour un RDV existant par id et rafraîchit (après UPDATE serveur)
supprimerRdv(id)	Supprime un RDV par id et rafraîchit (après DELETE serveur)
setVue(vue)	Change la vue — ‘jour’, ‘semaine’, ‘mois’, ‘annee’
allerA(dateStr)	Navigue vers une date ISO YYYY-MM-DD
getPeriode()	Retourne { debut, fin } de la période affichée

8. Vues

Vue jour

Affiche les créneaux d'une seule journée. Les RDV journée entière apparaissent dans un bandeau en haut. Clic sur l'en-tête d'un jour en vue semaine ou sur une case en vue année → bascule en vue jour.

Vue semaine

7 colonnes de Lundi à Dimanche avec les créneaux horaires. Clic sur l'en-tête d'un jour → vue jour de ce jour.

Vue mois

Grille classique. Maximum 3 badges par case, puis "+N autres". Double-clic sur une case → vue jour.

Vue année

12 mini-calendriers. Clic sur un nom de mois → vue mois. Clic sur un jour → vue jour. Les jours avec des RDV apparaissent en orange gras.

9. Drag & drop

Le drag & drop fonctionne dans les vues jour, semaine et mois. Glisser un badge RDV sur une autre case dispatche l'événement `acagenda:deplacer` avec l'ancienne et la nouvelle position. La mise à jour visuelle n'est pas automatique — c'est la page appelante qui appelle `setRdvs()` ou `modifierRdv()` après confirmation serveur.

 Le drag & drop ne modifie pas automatiquement les données locales — attendre la confirmation du backend avant de mettre à jour l'affichage.

10. Modale création / édition

La modale s'ouvre automatiquement au clic sur une case vide (création) ou sur un badge RDV (édition). Elle se ferme sur Echappée ou clic sur l'overlay.

Champs de la modale :

- Titre (requis)

- Date (requis, pré-remplie avec la date cliquée)
- Heure début / Heure fin (optionnelles, pré-remplies avec le créneau cliqué)
- Description (optionnelle)
- Palette de couleurs (10 pastilles cliquables)

En mode édition, un bouton Supprimer rouge apparaît à gauche du pied de modale.

11. callback_charger

La fonction `callback_charger` est appelée à chaque navigation avec les dates de début et de fin de la période affichée. Elle permet de charger les RDV depuis le serveur à la demande.

```
new ac_agenda('mon_agenda', {
  callback_charger: function(debut, fin) {
    fetch('/api/rdv?debut=' + debut + '&fin=' + fin)
      .then(r => r.json())
      .then(rdv => agenda.setRdvs(rdv));
  }
});
```

12. Bonnes pratiques

- Toujours attendre la réponse serveur avant d'appeler `ajouterRdv` / `modifierRdv` / `supprimerRdv`
- Utiliser `callback_charger` pour charger les RDV à la demande plutôt que de tout charger d'un coup
- Les ids des RDV doivent être uniques — utiliser les ids retournés par la base de données
- En mode `lecture_seule`, aucun clic ni drag & drop n'est possible — utile pour un affichage public
- La propriété canonique pour l'heure est `heure_debut` — éviter la propriété `heure` (compatibilité v1 uniquement)