ac_carte

Classe JS autonome — Carte Leaflet + OpenStreetMap avec géocodage

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

1. Présentation

ac_carte est une classe JavaScript autonome qui intègre une carte interactive Leaflet + OpenStreetMap dans n'importe quelle page. Elle gère le géocodage automatique d'adresses via Nominatim, un marqueur SVG coloré et un lien vers Google Maps pour le calcul d'itinéraire.

Nouveautés Mars 2026 :

- Clic sur la carte en mode saisie → reverse geocoding automatique (Nominatim)
- Adresse propre construite depuis les champs JSON Nominatim (house_number, road, postcode, city) — plus de display_name verbeux
- Numéro de rue conservé depuis la saisie brute si Nominatim ne le retourne pas
- Curseur crosshair sur la carte en mode saisie pour signaler la fonctionnalité

Caractéristiques principales :

- Classe JS unique — CSS embarqué, Leaflet chargé automatiquement via CDN
- Géocodage automatique : adresse texte → coordonnées GPS via Nominatim
- Reverse geocoding : clic sur la carte → adresse propre via Nominatim
- Adresse normalisée propre : "15 Rue de la Paix, 33000 Bordeaux"
- Initialisation par adresse ou par coordonnées GPS directes
- Marqueur SVG coloré personnalisable
- Popup avec l'adresse au clic sur le marqueur
- Lien "Calculer mon itinéraire" vers Google Maps
- Mode saisie avec géocodage automatique (délai 800ms)
- 100% gratuit — OpenStreetMap + Nominatim, zéro clé API
- Convention Classothèque — getValeur() / setValeur() pour intégration ac_grille

2. Installation et fichiers

Fichier	Emplacement
ac_carte.js	/assets/progiciel/classes/js/ac_carte.js

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

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

⚠️ *La classe charge Leaflet CSS et JS automatiquement via CDN au premier appel. Aucune autre dépendance n'est requise.*

3. Utilisation minimale

Dans le HTML, placer un div conteneur :

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

Par adresse (avec géocodage)

```
new ac_carte('ma_carte', {
    adresse : '12 rue de la Paix, Bordeaux'
});
```

Par coordonnées (plus rapide, sans appel réseau)

```
new ac_carte('ma_carte', {
    latitude  : 44.8404,
    longitude : -0.5805
});
```

4. Options de configuration

Option	Type	Défaut	Description
mode	string	'affichage'	'affichage' ou 'saisie'
adresse	string	null	Adresse à géocoder au chargement
latitude	number	null	Latitude GPS directe (prioritaire sur adresse)
longitude	number	null	Longitude GPS directe
zoom	int	15	Niveau de zoom Leaflet (1=monde, 19=rue)
hauteur	string	'350px'	Hauteur de la carte
largeur	string	'100%'	Largeur du composant
marqueur_couleur	string	'#5a7a62'	Couleur du marqueur SVG
lien_itineraire	bool	true	Afficher le bouton Calculer mon itinéraire
callback	function	null	fn(lat, lng, adresse) — appelée après positionnement

5. Géocodage Nominatim

Nominatim est le service de géocodage officiel d'OpenStreetMap. Il transforme une adresse texte en coordonnées GPS via une simple requête HTTP.

Caractéristique	Détail
Gratuit	100% gratuit, sans clé API
Limite	1 requête par seconde — largement suffisant pour un usage normal
Précision	Excellente pour les adresses françaises
Langue	Réponses en français (Accept-Language: fr)
URL search	https://nominatim.openstreetmap.org/search?addressdetails=1
URL reverse	https://nominatim.openstreetmap.org/reverse?addressdetails=1

Construction de l'adresse propre

La classe extrait les champs structurés de la réponse Nominatim pour construire une adresse courte et lisible :

```
// Champs utilisés dans address {}
house_number  → numéro de rue
road          → nom de rue (ou pedestrian, footway)
postcode      → code postal
city / town / village → commune


// Résultat
// '15 Rue de la Paix, 33000 Bordeaux'
// au lieu de 'Rue de la Paix, Bordeaux, Gironde, Nouvelle-Aquitaine, France
métropolitaine'
```

⚠️ *Si Nominatim ne retourne pas house_number, la classe extrait le numéro en tête de la saisie brute de l'utilisateur via une regex (gère bis, ter, quater).*

⚠️ *Si latitude et longitude sont fournis simultanément à adresse, les coordonnées sont prioritaires — Nominatim n'est pas appelé.*

6. Modes d'utilisation

6.1 Mode affichage

La carte s'affiche avec le marqueur positionné. Pas de champ de saisie. Idéal pour la page vitrine publique.

```
new ac_carte('carte_vitrine', {
    mode             : 'affichage',
    adresse          : '12 rue de la Paix, Bordeaux',
    marqueur_couleur : '#5a7a62',
    lien_itineraire  : true
});
```

6.2 Mode saisie

Un champ de saisie s'affiche au-dessus de la carte. Deux façons de positionner le marqueur :

- Saisie texte → géocodage automatique après 800ms ou au clic sur 🔍
- Clic sur la carte → reverse geocoding automatique → adresse propre dans le champ

Le curseur passe en crosshair sur la carte pour signaler que le clic est disponible.

```
new ac_carte('carte_saisie', {
    mode     : 'saisie',
    hauteur  : '300px',
    callback : function(lat, lng, adresse) {
        // Sauvegarder adresse en base
        console.log(adresse); // '15 Rue de la Paix, 33000 Bordeaux'
```

```
    }
});
```

7. Lien itinéraire

Quand lien_itineraire=true, un bouton "📍 Calculer mon itinéraire" s'affiche sous la carte après positionnement. Il ouvre Google Maps dans un nouvel onglet.

URL générée :

```
https://www.google.com/maps/dir/?api=1&destination={lat},{lng}
```

8. Méthodes publiques

Méthode	Retour	Description
setAdresse(adresse)	void	Géocode une adresse, centre la carte, met à jour le champ de saisie
setCoordonnees(lat, lng, adr)	void	Positionne directement par coordonnées. adr optionnel pour la popup.
getCoordonnees()	object	Retourne { lat, lng } de la position courante. null si aucune position.
getAdresse()	string	Retourne l'adresse propre courante (issue du géocodage ou fournie).
recentrer()	void	Recentre la carte sur le marqueur courant avec le zoom initial.
getValeur()	string	Alias Classothèque → getAdresse()
setValeur(v)	void	Alias Classothèque → setAdresse(v)

Exemple mode édition avec ac_grille

```
// La valeur est automatiquement injectée par ac_grille via setValeur()
// et récupérée via getValeur() à la sauvegarde.

// Configuration dans ac_grille :
['titre' => 'Adresse', 'champ' => 'adresse', 'type' => 'text',
    'widget'         => 'ac_carte',
    'widget_options' => ['mode' => 'saisie', 'hauteur' => '300px'],
],
```

9. Stockage en base de données

ac_carte retourne une adresse propre sous forme de chaîne. Un champ VARCHAR(300) est suffisant.

```
-- Colonne adresse simple
ALTER TABLE ma_table ADD COLUMN adresse VARCHAR(300) NULL;

-- Si vous souhaitez aussi stocker les coordonnées GPS
```

```
ALTER TABLE ma_table ADD COLUMN latitude  DECIMAL(10,7) NULL;
ALTER TABLE ma_table ADD COLUMN longitude DECIMAL(10,7) NULL;
```

⚠️ *Stocker latitude et longitude en DECIMAL(10,7) garantit une précision au centimètre près. Ne pas utiliser FLOAT qui peut introduire des erreurs d'arrondi.*

10. Intégration Classothèque (ac_grille)

ac_carte respecte la convention Classothèque — elle expose getValeur() et setValeur() utilisés par ac_grille pour l'initialisation et la récupération de la valeur.

Méthode	Alias de	Utilisée par ac_grille pour
getValeur()	getAdresse()	Récupérer l'adresse à sauvegarder en POST
setValeur(v)	setAdresse(v)	Initialiser le widget avec la valeur existante en BDD

⚠️ *ac_carte s'accroche à un (pas un <input>). Déclarer widget_ancre: 'div' dans la config colonne ac_grille — c'est la valeur par défaut donc ne rien préciser suffit.*

⚠️ *setValeur() est appelé avec un setTimeout(300ms) par ac_grille pour laisser Leaflet finir son initialisation asynchrone avant de géocoder l'adresse.*

11. Mode multi-marqueurs — setMarqueurs()

⚠️ **MODE NON TESTÉ —** *Cette fonctionnalité a été implémentée mais pas encore validée en conditions réelles. À tester avant mise en production.*

La méthode setMarqueurs() permet d'afficher plusieurs marqueurs sur une même carte. La carte ajuste automatiquement son zoom (fitBounds) pour les englober tous. Cas d'usage : carte de recherche de praticiens, clubs, mécanos…

Options de configuration spécifiques au mode multi

Option	Type	Défaut	Description
marqueur_couleur	string	'#5a7a62'	Couleur par défaut des marqueurs
couleur_selection	string	'#e8620a'	Couleur du marqueur sélectionné (orange artisan-code)
callback	function	null	fn(lat, lng, titre, marqueur) — appelée au clic sur un marqueur

Structure d'un marqueur

Propriété	Type	Requis	Description
lat	number	oui	Latitude GPS

Propriété	Type	Requis	Description
lng	number	oui	Longitude GPS
titre	string	non	Texte principal de la popup
description	string	non	Texte secondaire de la popup
couleur	string	non	Couleur SVG propre à ce marqueur (surcharge marqueur_couleur)

Exemple d'utilisation

```
const carte = new ac_carte('ma_carte', {
    hauteur          : '400px',
    marqueur_couleur : '#5a7a62',   // couleur par défaut
    couleur_selection: '#e8620a',   // orange au clic
    callback : function(lat, lng, titre, marqueur) {
        // Afficher la fiche du praticien, centrer un panneau…
        console.log('Sélectionné :', titre, marqueur);
    }
});

carte.setMarqueurs([
    { lat: 44.84, lng: -0.58, titre: 'Dr Dupont',  description:
'Kinésithérapeute' },
    { lat: 44.85, lng: -0.57, titre: 'Dr Martin',  description: 'Ostéopathe',
couleur: '#2563eb' },
    { lat: 44.83, lng: -0.59, titre: 'Dr Bernard', description: 'Acupuncteur' },
]);
```

Comportement

- La carte ajuste automatiquement le zoom pour que tous les marqueurs soient visibles (fitBounds avec padding 30px)
- Si un seul marqueur est fourni, la carte se centre dessus avec le zoom configuré
- Clic sur un marqueur → popup titre + description + callback fn(lat, lng, titre, marqueur)
- Le marqueur cliqué passe à couleur_selection, les autres reviennent à leur couleur d'origine
- Un appel à setMarqueurs() efface les marqueurs précédents
- setMarqueurs([]) vide la carte
- Chaque marqueur peut avoir sa propre couleur via la propriété couleur