

# Déployer & configurer le serveur de positions (@mostajs/net + data-plugin + orm sur SQLite)

**Auteur** : Dr Hamid MADANI [drmdh@msn.com](mailto:drmdh@msn.com) **Date** : 2026-06-05 **Cible** : amia ( `ssh amia` , user hmd) —  
exposé sous <https://geo.amia.fr/positions/> **Fichiers** : `deploy/ecosystem.positions.cjs` (PM2),  
`deploy/geo-positions.apache-snippet.conf` (vhost)

Met en place le **serveur de positions temps réel** qui alimente les dashboards ( `/cyclists/A/` ,  
`/cyclists/C/` ) en données **réelles** (SSE), au lieu du simulateur. Persistance **SQLite** via  
`@mostajs/data-plugin` → `@mostajs/orm` ; transport via `@mostajs/net` .

## 1. Architecture cible

```
device/tracker → @mostajs/net (REST write)      geo.amia.fr (Apache :443)
                |                               |
                ▼                               |— /           → PM2 geo (statique 4502)
                |                               |— /cyclists/  → statique (dashboards)
                |                               |— /positions/ → PM2 geo-positions (4610)
                ▼                               |
                @mostajs/data-plugin (MOSTA_DATA=orm)
                ▼
                @mostajs/orm (DB_DIALECT=sqlite) → data/cyclists.db
                ▲
dashboards ← SSE /positions/... (même origine, pas de CORS)
```

Deux process PM2 distincts : `geo` (site statique, 4502) et `geo-positions` (API temps réel, 4610).

## 2. Configuration (@mostajs/config)

Variables d'environnement (déjà câblées dans `deploy/ecosystem.positions.cjs` , ou via un `.env` )  
:

| Variable                          | Valeur                            | Rôle                                                        |
|-----------------------------------|-----------------------------------|-------------------------------------------------------------|
| <code>MOSTA_DATA</code>           | <code>orm</code>                  | data-plug : accès DB direct (pas de proxy net)              |
| <code>DB_DIALECT</code>           | <code>sqlite</code>               | orm : dialecte SQLite (driver <code>better-sqlite3</code> ) |
| <code>SGBD_URI</code>             | <code>.../data/cyclists.db</code> | chemin du <b>fichier SQLite</b> persistant                  |
| <code>MOSTA_NET_PORT</code>       | <code>4610</code>                 | port d'écoute du serveur net                                |
| <code>MOSTA_NET_TRANSPORTS</code> | <code>rest,sse,ws</code>          | transports activés (SSE pour le live)                       |
| <code>MOSTA_ENV</code>            | <code>PROD</code>                 | profil de config                                            |

Les noms exacts peuvent varier selon les versions ; recouper avec les `llms.txt` de `@mostajs/net`, `@mostajs/orm`, `@mostajs/data-plug`. Le couple `DB_DIALECT` + `SGBD_URI` est celui documenté par data-plug en mode `orm`.

## 3. Déploiement sur amia

### 3.1 Code + dépendances

```
# 1) pousser l'exemple vers un dossier dédié
rsync -az --exclude node_modules --exclude data \
  mosta-geo/examples/realtime-cyclists/ \
  amia:/home/hmd/prod/geo-positions/

# 2) installer les dépendances (SQLite natif inclus)
ssh amia 'cd /home/hmd/prod/geo-positions && \
  npm init -y >/dev/null 2>&1; \
  npm i @mostajs/config @mostajs/data-plug @mostajs/orm @mostajs/net @mostajs/geo better-sqlite3 \
  && \
  mkdir -p data logs'
```

### 3.2 Process PM2

```
ssh amia 'cd /home/hmd/prod/geo-positions && \
  pm2 start deploy/ecosystem.positions.cjs && pm2 save'
# vérifier le port local :
ssh amia 'curl -s -o /dev/null -w "%{http_code}\n" http://127.0.0.1:4610/'
```

La première écriture crée et migre `data/cyclists.db` (stratégie de schéma orm). SQLite ne demande aucun serveur DB — un simple fichier.

### 3.3 Exposition Apache (sous /positions/)

Insérer le snippet `deploy/geo-positions.apache-snippet.conf` dans le vhost :443 de `geo.amia.fr` avant le `ProxyPass /` existant, puis :

```
ssh amia 'sudo a2enmod proxy proxy_http proxy_wstunnel rewrite headers && \
sudo apache2ctl configtest && sudo systemctl reload apache2'
```

(`sudo` requiert le mot de passe.) Test :

```
curl -s -o /dev/null -w "%{http_code}\n" https://geo.amia.fr/positions/
```

## 4. Alimenter en positions

- **Écriture** : un device/tracker POSTe les positions via REST de net ( `POST https://geo.amia.fr/positions/Position` ), ou le `tracker.mjs` / `gateway.mjs` écrit via `data-plugin` ( `db.create('Position', ...)` ).
- **Lecture live** : les dashboards s'abonnent au **SSE** de net sous `/positions/` (même origine que `/cyclists/` → pas de CORS).

Pour brancher les dashboards sur le vrai flux, remplacer le simulateur ( `setInterval(tick,...)` ) par une source `EventSource('/positions/...')` (SSE) qui appelle `computeGeo()` sur chaque position reçue. Le chemin SSE exact dépend de la config de `@mostajs/net` (cf. son `llms.txt`).

## 5. SQLite — exploitation

- **Fichier** : `data/cyclists.db` (+ `-wal` / `-shm` en mode WAL). Sauvegarde = copie du fichier.
- **Sauvegarde à chaud** : `ssh amia 'sqlite3 /home/hmd/prod/geo-positions/data/cyclists.db ".backup /home/hmd/backups/cyclists-$(date +%F).db"'`.
- **Montée en charge** : SQLite convient à un événement ponctuel ; au-delà, basculer le **même code** vers PostgreSQL/PostGIS en changeant juste `DB_DIALECT` / `SGBD_URI` (force de `@mostajs/orm` — zéro réécriture applicative).

## 6. Mise à jour

---

```
rsync -az --exclude node_modules --exclude data --exclude logs \
  mosta-geo/examples/realtime-cyclists/ amia:/home/hmd/prod/geo-positions/
ssh amia 'pm2 restart geo-positions'
```