

Mise à jour des dépendances

Mise à jour des dépendances

📅 Dernière mise à jour : 2026-05-10

Politique : manuel

Pas de Renovate / Dependabot configurés. Mises à jour manuelles, en bloc, quand tu en as envie.

Pourquoi pas Renovate ?

- Volume de PR auto trop élevé sur un projet d'un seul mainteneur
- Préférence pour batch les mises à jour (1× / 2-3 mois)
- Évite la friction "PR Renovate qui casse la CI le mardi matin"

Si tu changes d'avis : Renovate self-hosted via Docker, conf simple `renovate.json` à la racine.

Workflow de batch update

Backend

```
# 1. Voir les outdated
npm outdated

# 2. Patch updates (safe – bug fixes)
npm update
```

```
# 3. Minor / major sélectifs
npm install hono@latest
npm install drizzle-orm@latest
npm install zod@latest
npm install @types/node@latest

# 4. Vérifier
npm run build
npm test

# 5. Tester en dev
npm run dev
# Aller sur /play, poser quelques questions, vérifier RAG OK

# 6. Si tout OK : commit + push → CI build + push image → pull en prod
git add package.json package-lock.json
git commit -m "chore(deps): bump backend deps"
git push
```

Frontend

```
cd frontend
npm outdated
npm update

# Major upgrades
npm install vue@latest vue-router@latest pinia@latest
npm install tailwindcss@latest @tailwindcss/vite@latest
npm install marked@latest

npm run build
npm test
npm run dev
# Tester l'UI

cd ..
git add frontend/package.json frontend/package-lock.json
```

```
git commit -m "chore(deps): bump frontend deps"
git push
```

Cas spéciaux

`@flesh-and-blood/cards` + `@flesh-and-blood/types`

Ces deux packages contiennent les **données cartes FAB**. Mettre à jour = nouveau set FAB.
Workflow détaillé : `tcg-integrations/mettre-a-jour-cartes.md`.

```
npm update @flesh-and-blood/cards @flesh-and-blood/types
# Build + push obligatoire pour que le container voie les nouvelles cartes
```

`mana-font` (frontend)

Webfont Andrew Gioia pour symboles MTG. Update rare (pas de nouveau symbole MTG depuis longtemps). Si nouvelle release :

```
cd frontend
npm install mana-font@latest
# Vérifier visuellement qu'aucun symbole existant ne casse
```

`pdfjs-dist`

Couche d'extraction PDF. Updates régulières mais souvent breaking. Tester sur quelques PDFs de jeux différents avant de push.

`better-sqlite3`

Driver SQLite synchrone. Update prudemment — il y a souvent des prebuilds Node-version-specific. Si update casse en CI :

- Vérifier compat Node 22
- Si Alpine cible : prebuild musl absent → bloque tout, voir `pipeline-cicd.md` pour le contexte

Audit de sécurité

```
npm audit  
cd frontend && npm audit
```

Souvent du bruit (vulnérabilités CVSS bas dans des transitive deps). Filtrer :

```
npm audit --audit-level=high
```

Update Node 22 → 23 / 24

Pas urgent. Mais quand tu décides :

1. Modifier `engines.node` dans `package.json`
2. Modifier le base image dans `Dockerfile` (`FROM node:24-bookworm-slim`)
3. Modifier le base image dans `.gitea/workflows/build.yml` (container test)
4. `npm ci` localement avec Node 24, vérifier que tout compile / test

Pas de major version pinning

`package.json` utilise `^` partout (= compatible minor). Pas de pinning strict. Si tu veux pin une version pour reproduire un bug :

```
npm install hono@4.7.6 --save-exact
```

`--save-exact` retire le `^`.

Revision #1

Created 2026-05-10 15:20:10 UTC by thymon

Updated 2026-05-10 15:20:10 UTC by thymon