

# ADR-001 : Hono plutôt qu'Express ou Fastify

# ADR-001 : Hono plutôt qu'Express ou Fastify

“ Date : 2026 (initial setup) — Statut : accepté

## Contexte

Au moment de choisir le framework backend pour un projet TypeScript moderne avec :

- Streaming SSE intensif (RAG)
- Validation typée (Zod inline)
- Self-hosting (image Docker légère)
- Single dev (peu de bandwidth pour gérer un écosystème complexe)

Les options évaluées : Express (mature mais legacy), Fastify (perf++ mais plugins gros), Hono (moderne, types natifs, perf comparable Fastify).

## Décision

**Hono**, version 4.x.

Raisons principales :

- **SSE natif** via `streamSSE()` helper. Express demande des wrappers tiers, plus de friction.

- **Types end-to-end** : context typé (`AppEnv`), middleware typé, pas de `any` parasite.
- **Légèreté** : core ~10 KB, pas de plugins lourds par défaut. Image Docker plus petite.
- **Performance** : benchmarks proches de Fastify, ~2-3× Express. Suffisant pour RAG self-hosted.
- **Ergonomie** : routes composables, middleware factorisable, validation Zod naturelle.

# Conséquences

## Bonnes

- Le code est concis et bien typé.
- SSE marche out-of-the-box avec heartbeat helpers.
- Migrations entre versions Hono peu invasives.

## Mauvaises

- Communauté plus petite qu'Express : moins de tutoriels, moins de StackOverflow.
- Pas (ou peu) de plugins prêts à l'emploi pour des trucs ésotériques. Mais on n'en a pas besoin.
- Si un jour besoin de scaler horizontalement, le pattern `streamSSE` + heartbeats est portable mais demande de la doc maison.

# Alternative envisagée

- **Fastify** : aussi rapide, plus mature. Mais l'écosystème plugin est plus complexe (lifecycle hooks, fastify-types, etc.) — plus de friction pour un single dev.
- **Express** : trop legacy. Pas de types natifs, SSE = wrappers, validation = Joi/Zod custom. Peu de plaisir à coder en TS.

---

Revision #1

Created 2026-05-10 15:19:50 UTC by thymon

Updated 2026-05-10 15:19:50 UTC by thymon