

KoaJS, HapiJS, Fastify, Django Ninja

Note: Any backend framework that supports OAuth2 (any one used in production will support this) can work with Keycloak. Sometimes there will be built in tools to work with keycloak, such as an adapter, but they are not strictly necessary.

Koa.js

- Lost traction despite having better features than Express
- Community support is lacking
- Very slow to no active development. Latest release has taken over a year to complete.

HapiJS

- Small community
- High speed
- Maintainer stepped down due to lack of financial support and handed off development to the community. There have been no substantial changes since 2022.

<https://github.com/hapijs/hapi/issues/4111>

Fastify

- Potential code gymnastics to get TS working smoothly. Fastify is written in vanilla JS so they have to manually keep all the type definitions up to date. The framework wasn't necessarily built with TS in mind. That doesn't mean that TS doesn't work or isn't supported. But it may be a pain point.
- Very high speed. Potentially the fastest backend application framework. Express can take up to double the amount of time to process the same number of requests. Even though Fastify is ridiculously fast, that doesn't mean that Express isn't fast enough. Fastify's speed increase may be negligible considering we're not dealing with 10's of thousands of requests simultaneously.
- Route / Request validation through JSON schemas
- Pretty robust documentation / community. Not quite Express size, but still sizable and growing.
- Plugin ecosystem system for adding functionality. Pretty much everything we'd need would be covered here.

Django Ninja

- No Prisma support, but does integrate with the Django ORM
- Would pull us into the Django ecosystem. (Plugins / extensions, ORM, authentication)
- Slower than the other options. But still probably fast enough
- Simple syntax similar to that of FastAPI
- More of an API building tool than a backend application framework. It's focus is more on content delivery through API routes instead of handling business logic, keeping track of users, and updating a database.