

Hono Bun Native Project Structure

Hono optimized for Bun's native runtime. Uses Bun's built-in SQLite, file APIs, and fast HTTP server.

#hono #typescript #bun #sqlite #native #performance

PNG

PDF

Copy

</> Prompt

Project Directory

myproject/

- >  **src/**
-  index.ts

Bun.serve entry
-  app.ts

Hono app
- >

 **routes/**
-  index.ts
-  users.ts
-  files.ts
- >

 **services/**
-  user.service.ts
-  file.service.ts

Bun.file APIs
- >

 **db/**
-  index.ts

bun:sqlite clie...
-  schema.sql
-  **migrations/**
- >

 **middleware/**
-  auth.ts
-  error-handler.ts
- >

 **lib/**
-  env.ts
-  password.ts

Bun.password
- >

 **tests/**

Bun test runner
-  users.test.ts
-  setup.ts
-  **data/**

SQLite database...
-  package.json
-  tsconfig.json
-  bunfig.toml

Bun configurati...
-  .env
-  .gitignore

Why This Structure?

Bun's native APIs are faster than Node equivalents. Built-in SQLite via `bun:sqlite`, fast file I/O with `Bun.file()`, and native password hashing with `Bun.password`. Hono runs on Bun's HTTP server with zero adapter overhead.

Key Directories

- src/db/** - Uses `bun:sqlite` for zero-dependency SQLite
- src/lib/password.ts** - Native `Bun.password` for hashing
- src/services/file.service.ts** - Fast file ops with `Bun.file()`
- data/** - SQLite database files, gitignored

Getting Started

- `bun create hono my-app`
- `bun install`
- `bun run src/index.ts`
- Database auto-creates in `data/`

Bun SQLite

```
// src/db/index.ts
import { Database } from 'bun:sqlite';

const db = new Database('data/app.db');
db.exec('PRAGMA journal_mode = WAL;');

export { db };
```

When To Use This

- Maximum performance is priority
- Self-hosted or VPS deployment
- Need built-in SQLite without dependencies
- Prefer Bun's tooling over Node
- Single-server or container deployment

Trade-offs

- Not edge** - Bun runs on servers, not edge runtimes
- Bun maturity** - Newer runtime, some Node APIs missing
- Ecosystem** - Some npm packages may not work

Best Practices

- Use WAL mode for SQLite concurrency
- Leverage `Bun.password` over bcrypt
- Use `Bun.file()` for streaming large files
- Run tests with `bun test` —it's built in
- Use `bunfig.toml` for Bun-specific config