Other install methods  ›  Bun (Experimental)

Other install methods

# Bun (Experimental)

Goal: run this repo with **Bun** (optional, not recommended for WhatsApp/Telegram) without diverging from pnpm workflows.

⚠️ **Not recommended for Gateway runtime** (WhatsApp/Telegram bugs). Use Node for production.

## Status

Bun is an optional local runtime for running TypeScript directly (`bun run …`, `bun --watch …`).

`pnpm` is the default for builds and remains fully supported (and used by some docs tooling).

Bun cannot use `pnpm-lock.yaml` and will ignore it.

## Install

Default:

```
bun install
```

Note: `bun.lock` / `bun.lockb` are gitignored, so there's no repo churn either way. If you want *no lockfile writes*:

```
bun install --no-save
```

>

# Build / Test (Bun)

```
bun run build
bun run vitest run
```

## Bun lifecycle scripts (blocked by default)

Bun may block dependency lifecycle scripts unless explicitly trusted
( `bun pm untrusted` / `bun pm trust` ). For this repo, the commonly blocked
scripts are not required:

  `@whiskeysockets/baileys` `preinstall` : checks Node major $\geq$ 20 (we run
  Node 22+).

  `protobufjs` `postinstall` : emits warnings about incompatible version
  schemes (no build artifacts).

If you hit a real runtime issue that requires these scripts, trust
them explicitly:

```
bun pm trust @whiskeysockets/baileys protobufjs
```

## Caveats

  Some scripts still hardcode pnpm (e.g. `docs:build` , `ui:*` ,
  `protocol:check` ). Run those via pnpm for now.

Powered by mintlify

›