



# Typebot Project Structure

The visual chatbot builder. A comprehensive monorepo featuring a Next.js builder, a separate viewer application, and extensive shared packages.

Updated 2025-12-30

#typebot #nextjs #monorepo #turborepo #chatbot #open-source

PNG

PDF

Copy

Prompt

## Project Directory

### typebot.io/

- apps/ Main applicatio...
- builder/ The main dashbo...
  - src/ Application sou...
  - public/ Static assets
  - next.config.mjs
  - viewer/ Lightweight cha...
  - landing-page/ Marketing site
  - docs/ Documentation (...)
- packages/ 40+ shared pack...
  - bot-engine/ Core execution ...
  - schemas/ Zod schemas & t...
  - prisma/ Database client...
  - ui/ Shared Chakra U...
  - embeds/ JS embed librar...
  - ... 35+ other packa...
- turbo.json Turborepo pipel...
- package.json Root workspace
- docker-compose.yml Self-hosting se...

## Repository Info

Repository - [baptisteArno/typebot.io](#)

Stars - 15k+

License - AGPL-3.0

Last Analyzed - December 2025

## Tech Stack

Framework - Next.js (Pages Router)

Language - TypeScript

Database - PostgreSQL with Prisma

UI Library - Chakra UI

Monorepo - Turborepo + pnpm

State - Zustand

## Architecture Notes

Typebot separates the 'builder' (where users create bots) from the 'viewer' (where bots run). This allows the viewer to be extremely lightweight and fast. The `bot-engine` package is the heart of the system, shared by both apps. They use a very granular package strategy with over 40 packages for specific domains like `ai`, `billing`, and `blocks`.

## Key Directories

**apps/builder/** - The admin dashboard for creating flows

**apps/viewer/** - The runtime that displays the chat to end users

**packages/bot-engine/** - The core logic for traversing the chat graph

**packages/embeds/** - Scripts for embedding Typebot on other sites (React, vanilla JS)

## Why This Structure?

A perfect example of a complex, feature-rich React application broken down into manageable pieces. The separation of `builder` and `viewer` is a classic pattern for tools that produce user-facing content.