



# ToolJet Project Structure

The open-source low-code framework for building business applications. A robust TypeScript application built with NestJS, React, and a modular plugin system.

Updated 2025-12-30

#tooljet #typescript #nestjs #react #low-code #monorepo #plugin-architecture

PNG

PDF

Copy

Prompt

### Project Directory

ToolJet/

- server/ NestJS API Serv...
  - src/modules/ Core API logic
  - src/entities/ TypeORM models
  - migrations/
- frontend/ React Builder UI
  - src/AppBuilder/ The main editor
  - src/Editor/ Editor componen...
- plugins/ Modular Connect...
  - packages/ Official integr...
- cli/ ToolJet Command...
- deploy/ K8s, Docker, Te...
- package.json

### Repository Info

**Repository** - ToolJet/ToolJet

**Stars** - 25k+

**License** - AGPL-3.0

**Last Analyzed** - December 2025

### Tech Stack

**Language** - TypeScript

**Backend** - Node.js (NestJS)

**Frontend** - React

**Database** - PostgreSQL

**ORM** - TypeORM

**Infrastructure** - Terraform / Kubernetes

### Architecture Notes

ToolJet follows a classic Enterprise TypeScript architecture. The backend is a structured **NestJS** application using **TypeORM** for robust database interactions. The frontend is a massive React SPA that handles the complex state of the application builder. A key architectural pillar is the **plugins/** directory, which provides a standardized way to add third-party integrations (connectors) without modifying the core server code.

### Key Directories

**server/src/modules/** - Contains the core features of the platform, including workspace management, authentication, and the query execution engine.

**plugins/packages/** - Where the 'Connectors' live. Each sub-folder is a separate package for a specific data source (e.g., **postgresql**, **googlesheets**, **restapi**).

**frontend/src/AppBuilder/** - The most complex part of the frontend. It contains the logic for the visual editor, including drag-and-drop, component rendering, and the inspector panel.

### Why This Structure?

ToolJet is a great example of a 'Pluggable Monolith'. While the core is a single system, the plugin architecture allows it to scale horizontally in terms of features (integrations) while remaining maintainable.