



FastAPI Minimal Project Structure

Single file API structure for quick prototyping. Ideal for learning, small APIs, and rapid development.

Updated 2025-12-22

#fastapi #python #api #rest #minimal

PNG

PDF

Copy

Prompt

Project Directory



myproject/

- main.py All routes in o...
- requirements.txt
- .env.example Copy to .env
- .gitignore
- README.md

Why This Structure?

This is the simplest possible FastAPI structure—everything in `main.py`. No complex imports, no module hunting. Perfect for prototypes, tutorials, or internal tools with < 10 endpoints. Start here and grow organically.

Key Directories

- main.py** - App instance, routes, models all in one place
- .env.example** - Environment variables template

Getting Started

- `python -m venv venv && source venv/bin/activate`
- `pip install -r requirements.txt`
- `cp .env.example .env`
- `uvicorn main:app --reload`
- Visit `http://localhost:8000/docs` for Swagger UI

When To Use This

- Learning FastAPI fundamentals
- Quick prototypes and proof-of-concepts
- Internal tools with few endpoints
- Hackathons and time-constrained projects
- Single-developer side projects

When To Upgrade

- `main.py` exceeds 200 lines
- You need database models (move to `async-sqlalchemy`)
- Multiple developers working simultaneously
- Need for proper testing structure
- Deployment requirements emerge

Trade-offs

- No separation** - Everything in one file becomes hard to navigate
- No tests folder** - Add tests/ when needed, but not premature
- Limited scalability** - By design—upgrade when you outgrow it