Spring Boot Minimal Project Structure

Single module structure with basic REST controller. Ideal for learning Spring Boot, quick prototypes, and small APIs.

#spring-boot #java #rest #api #minimal

PNG PDF Copy Prompt

Project Directory

```
myproject/
  src/
    main/
      java/
        com/example/demo/  Base package
          DemoApplication.java  @SpringBootAppl…
          HelloController.java  @RestController
      resources/
        application.properties  Or .yml
        static/  Static assets
        templates/  Thymeleaf templ…
    test/
      java/
        com/example/demo/
          DemoApplicationTests.java  @SpringBootTest
  pom.xml  Maven build
  .gitignore
  README.md
```

Why This Structure?

This is Spring Initializr's default structure—everything in one package. The `@SpringBootApplication` class bootstraps the app with auto-configuration. Perfect for understanding Spring Boot fundamentals before adding layers.

Key Directories

`src/main/java/` - All Java source code, organized by package

`src/main/resources/` - Configuration files, static assets, templates

`application.properties` - Server port, database URL, and all configuration

Getting Started

1. `./mvnw spring-boot:run` (or `mvn spring-boot:run`)
2. Visit http://localhost:8080
3. Edit `application.properties` for configuration

When To Use This

- Learning Spring Boot patterns
- Quick prototypes and MVPs
- Simple microservices with few endpoints
- Internal tools and utilities
- Proof of concept implementations

When To Upgrade

- More than 5-10 endpoints needed
- Database access required
- Multiple developers on the project
- Need separation of concerns
- Business logic becoming complex

Trade-offs

`No layers` - Controller handles everything—add services when logic grows

`No database` - Add Spring Data JPA when persistence is needed

`Flat package` - All classes in one package—fine for small projects