

index of all express js topics that all professional backend engineer should know

Here's a **practical, professional-level index of Express.js topics** a strong backend engineer is expected to know. It's organized from fundamentals → advanced → production-level concerns so you can use it as a roadmap or checklist.

≡ ChatGPT 5.2 ▾

🔄 Temporary Chat ...

✓ 1) Core Fundamentals

Node.js Foundations (Prerequisite)

- Event loop & async model
- CommonJS vs ES Modules
- Streams & buffers
- Environment variables
- File system basics

Express Basics

- What is Express & how it works
- Creating a server
- Routing basics (GET, POST, PUT, DELETE)
- Request & Response objects
- Route parameters & query strings
- Express application structure

✓ 2) Routing & Middleware (Very Important)

Routing

- Router module (`express.Router`)
- Route grouping & modular routes

- Route chaining
- Dynamic routes
- Regex routes

Middleware

- What middleware is
 - Application vs router middleware
 - Built-in middleware (`express.json` , `express.static`)
 - Third-party middleware
 - Custom middleware
 - Error-handling middleware
 - Middleware order & execution flow
-

✓ 3) Request Handling

- Handling form data
 - JSON payload handling
 - File uploads (Multer)
 - Headers & cookies
 - CORS handling
 - Validation (Joi, Zod, express-validator)
-

✓ 4) Templating & Views (Optional but useful)

- Template engines (EJS, Pug, Handlebars)
 - Rendering dynamic views
 - Layouts & partials
-

✓ 5) REST API Design

- REST principles
- Resource naming conventions
- Versioning APIs

- Pagination & filtering
 - Status codes best practices
 - Idempotency
 - HATEOAS (optional advanced)
-

✓ 6) Databases Integration

SQL

- Connecting with PostgreSQL/MySQL
- ORMs (Sequelize, Prisma, TypeORM)
- Migrations & schema design

NoSQL

- MongoDB integration
 - ODM (Mongoose)
 - Indexing & performance
-

✓ 7) Authentication & Authorization

- Sessions vs JWT
 - OAuth basics
 - Passport.js
 - Role-based access control (RBAC)
 - Refresh tokens
 - Secure password hashing (bcrypt/argon2)
-

✓ 8) Security Best Practices (Critical for pros)

- Helmet.js
- CORS configuration
- Rate limiting
- XSS & CSRF protection
- Input sanitization

- Preventing SQL/NoSQL injection
 - Secure cookies
 - HTTPS & TLS
 - Environment variable management
-

✓ 9) Error Handling & Logging

- Centralized error handling
 - Custom error classes
 - Structured logging
 - Winston / Pino logging
 - Handling uncaught exceptions & rejections
-

✓ 10) Performance & Scalability

- Compression middleware
 - Caching strategies (Redis)
 - Load balancing
 - Clustering (Node cluster/PM2)
 - Horizontal scaling
 - Memory leak detection
-

✓ 11) Testing

- Unit testing (Jest, Mocha)
 - Integration testing
 - API testing (Supertest)
 - Mocking databases
 - Test coverage
-

✓ 12) Production & DevOps

- Environment configs

- Dockerizing Express apps
 - CI/CD pipelines
 - Reverse proxies (Nginx)
 - Process managers (PM2)
 - Monitoring & health checks
-

13) Advanced Concepts

- Microservices architecture
 - API Gateway pattern
 - WebSockets (Socket.io)
 - GraphQL with Express
 - Background jobs & queues (BullMQ)
 - Event-driven architecture
-

14) Code Quality & Architecture

- MVC architecture
 - Clean architecture principles
 - Dependency injection
 - Repository pattern
 - Service layer pattern
 - Modular folder structure
-

15) Documentation

- Swagger / OpenAPI
 - Postman documentation
 - API version documentation
-

Pro-Level Bonus Topics

- Multi-tenancy

- Internationalization (i18n)
 - Feature flags
 - Rate limiting strategies
 - Serverless Express
 - Observability (metrics, tracing)
-

If you want, I can next give:

- ✓ A learning roadmap (beginner → job-ready)
- ✓ A project-based roadmap
- ✓ A backend engineer interview checklist
- ✓ A “Top 20% topics that get you hired fastest” list

Just tell me 👍

