

BASIC LEVEL (Getting Started)

Learn the fundamentals first.

1. What is Express.js

- Role of Express in Node.js
- Installing Express (npm install express)
- Hello World Example

2. Basic Server Setup

- Creating a simple server
- Listening to a port
- Sending responses (res.send, res.json, res.sendFile)

3. Routing Basics

- app.get(), app.post(), app.put(), app.delete()
- Route parameters (req.params)
- Query parameters (req.query)
- Route grouping

4. Middleware Basics

- What is middleware?
- app.use() and next()
- Built-in middleware (express.json(), express.urlencoded())
- Custom middleware (logging, authentication placeholders)

5. Serving Static Files

- express.static() for HTML, CSS, JS, Images
- Folder structure for static assets

6. Handling Forms & JSON Data

- POST requests using express.urlencoded()
- Handling JSON with express.json()

□ INTERMEDIATE LEVEL (Practical Concepts)

Once you understand basics, move on to real project concepts.

7. Templating Engines

- Using EJS, Pug, or Handlebars
- Passing data to templates

8. Modularizing Routes

- Using express.Router()
- Organizing routes in separate files

9. Error Handling

- Default error handler
- Custom error middleware
- 404 Not Found handler

10. Environment Variables

- Using .env and dotenv package
- Accessing process.env

11. Working with Databases

- MongoDB connection with mongoose
- Basic CRUD operations
- Handling async/await with Express

12. RESTful APIs

- REST API design principles
- CRUD endpoints (GET, POST, PUT, DELETE)
- Postman testing

13. CORS (Cross-Origin Resource Sharing)

- Understanding CORS
- Using the cors middleware

14. File Uploads

- Using multer for handling uploads

15. Session & Cookies

- express-session and cookie-parser
 - Basic authentication using sessions
-



ADVANCED LEVEL (Real-World Topics)

16. JWT Authentication

- Using jsonwebtoken
- Protecting routes with tokens
- Refresh tokens concept

17. Advanced Middleware

- Role-based access control (RBAC)
- Rate limiting
- Helmet for security headers

18. Error Logging & Debugging

- Using morgan for logs
- Custom loggers (e.g., Winston)
- Handling async errors

19. API Validation

- Using express-validator or joi
- Validating request body, params, and queries

20. Versioning APIs

- Organizing v1, v2 APIs
- Deprecating old versions

21. Pagination & Filtering

- Query-based pagination
- Sorting and filtering with MongoDB

22. File System Operations

- Reading/writing files with fs module
- Serving dynamic files

23. Deploying Express Apps

- Hosting on Render, Vercel, or Railway
- Environment setup for production
- Using pm2 to keep server running

24. WebSockets (Optional Advanced)

- Real-time communication with socket.io

25. Testing Express Apps

- Using mocha or jest
- Testing routes and middleware



BONUS (Best Practices & Patterns)

- Folder structure for large Express apps
- MVC architecture
- Using async/await with try/catch
- Writing clean & reusable middleware
- Using TypeScript with Express