

# Express.js Interview — 50 HighYield Q&A; (with quick answers & examples)

This document provides 50 Express.js interview-style questions with concise answers and code snippets/examples.

## Q1. What is Express.js?

**Quick answer:** Minimal Node.js framework for building web apps and APIs.

**Example:** `const express=require('express'); const app=express();`

## Q2. How do you create a simple Express server?

**Quick answer:** Use `express()`, define routes, listen on port.

**Example:** `const app=express(); app.get('/',(req,res)=>res.send('Hi')); app.listen(3000);`

## Q3. What are middleware functions?

**Quick answer:** Functions that process req/res before route handler.

**Example:** `app.use((req,res,next)=>{console.log('req'); next();});`

## Q4. Difference between `app.use` and `app.get`?

**Quick answer:** `app.use`: middleware for all methods/paths. `app.get`: specific route handler.

**Example:** `app.use(auth); app.get('/home',fn);`

## Q5. What is error-handling middleware?

**Quick answer:** Middleware with 4 args (`err, req, res, next`).

**Example:** `app.use((err,req,res,next)=>{res.status(500).send(err.message);});`

## Q6. How do you handle 404 errors?

**Quick answer:** Define catch-all route after all handlers.

**Example:** `app.use((req,res)=>res.status(404).send('Not Found'));`

## Q7. What is `req.params`?

**Quick answer:** Object with route parameters.

**Example:** `app.get('/user/:id',(req,res)=>res.send(req.params.id));`

## Q8. What is `req.query`?

**Quick answer:** Object with URL query string parameters.

**Example:** `/search?q=book => req.query.q`

## Q9. What is req.body?

**Quick answer:** Object containing parsed request body (requires body-parser/json middleware).

**Example:** `app.use(express.json());`

## Q10. How do you serve static files?

**Quick answer:** Use `express.static` middleware.

**Example:** `app.use(express.static('public'));`

## Q11. Difference between res.send and res.json?

**Quick answer:** `send`: sends any type. `json`: sends JSON with proper header.

**Example:** `res.json({msg:'ok'});`

## Q12. How to redirect in Express?

**Quick answer:** Use `res.redirect` with URL.

**Example:** `res.redirect('/login');`

## Q13. How do you set response headers?

**Quick answer:** Use `res.set` or `res.header`.

**Example:** `res.set('Content-Type','text/plain');`

## Q14. How do you chain middleware?

**Quick answer:** Pass control via `next()` to next middleware.

**Example:** `app.use(auth); app.use(logger);`

## Q15. What is next() in middleware?

**Quick answer:** Passes control to next middleware function.

**Example:** `next();`

## Q16. Difference between router and app in Express?

**Quick answer:** Router is modular route handler, app is main server.

**Example:** `const router=express.Router(); router.get('/',fn); app.use('/api',router);`

## Q17. How to handle file uploads?

**Quick answer:** Use `multer` middleware.

**Example:** `const upload=require('multer')(); app.post('/upload',upload.single('f'),fn);`

## Q18. What is helmet in Express?

**Quick answer:** Security middleware setting HTTP headers.

**Example:** `app.use(require('helmet')());`

## Q19. How to enable CORS?

**Quick answer:** Use cors middleware package.

**Example:** `const cors=require('cors'); app.use(cors());`

## Q20. How to create RESTful routes in Express?

**Quick answer:** Use HTTP verbs GET, POST, PUT, DELETE for resources.

**Example:** `app.post('/users',fn); app.put('/users/:id',fn);`

## Q21. How do you handle sessions?

**Quick answer:** Use express-session middleware.

**Example:** `app.use(session({secret:'x'}));`

## Q22. What is cookie-parser?

**Quick answer:** Middleware to parse cookies.

**Example:** `app.use(require('cookie-parser')());`

## Q23. How do you log requests?

**Quick answer:** Use morgan middleware.

**Example:** `app.use(require('morgan')('dev'));`

## Q24. What is difference between PUT and PATCH routes?

**Quick answer:** PUT replaces resource. PATCH partially updates.

**Example:** `app.put('/user/:id',fn);`

## Q25. What is app.all()?

**Quick answer:** Handles all HTTP methods for given path.

**Example:** `app.all('/test',(req,res)=>res.send('All methods'));`

## Q26. What is difference between app.listen and http.createServer?

**Quick answer:** app.listen is shortcut for http.createServer with Express app.

**Example:** `const server=http.createServer(app);`

## Q27. How do you modularize routes?

**Quick answer:** Use express.Router in separate files and mount with app.use.

**Example:** `app.use('/users',require('./routes/users'));`

### **Q28. What is res.locals?**

**Quick answer:** An object to store response-local variables accessible in templates.

**Example:** `res.locals.user=user; next();`

### **Q29. What are template engines with Express?**

**Quick answer:** Engines like EJS, Pug render dynamic HTML.

**Example:** `app.set('view engine','ejs');`

### **Q30. How to parse URL-encoded data?**

**Quick answer:** Use `express.urlencoded` middleware.

**Example:** `app.use(express.urlencoded({extended:true}));`

### **Q31. What is difference between synchronous and asynchronous middleware?**

**Quick answer:** Sync executes immediately; async returns promise/calls next in callback.

**Example:** `app.use(async(req,res,next)=>{await db(); next();});`

### **Q32. What is express-generator?**

**Quick answer:** CLI tool to scaffold Express apps.

**Example:** `npx express-generator`

### **Q33. What is app.route()??**

**Quick answer:** Chain route handlers for same path.

**Example:** `app.route('/book').get(fn).post(fn);`

### **Q34. How do you set environment variables in Express?**

**Quick answer:** Use `process.env` or `dotenv` package.

**Example:** `const PORT=process.env.PORT;`

### **Q35. How to handle JSON parse errors?**

**Quick answer:** Error-handling middleware catches bad JSON.

**Example:** `app.use((err,req,res,next)=>{ if(err.type==='entity.parse.failed'){res.status(400).send('Invalid JSON');}});`

### **Q36. What is middleware order importance?**

**Quick answer:** Runs sequentially; wrong order may break logic.

**Example:** `app.use(auth); app.use(routes);`

### **Q37. How do you disable x-powered-by header?**

**Quick answer:** `app.disable('x-powered-by');`

**Example:** Example usage or snippet here.

### **Q38. How do you trust proxy in Express?**

**Quick answer:** Enable when behind reverse proxy.

**Example:** `app.set('trust proxy',1);`

### **Q39. What is `res.download()`?**

**Quick answer:** Transfers file as attachment.

**Example:** `res.download('/path/file.pdf');`

### **Q40. What is `res.sendFile()`?**

**Quick answer:** Sends static file as response.

**Example:** `res.sendFile(__dirname+'/index.html');`

### **Q41. What is difference between `res.end` and `res.send`?**

**Quick answer:** `end`: ends response without data. `send`: ends with data.

**Example:** `res.end();`

### **Q42. How do you return status codes?**

**Quick answer:** `res.status(201).send('Created');`

**Example:** Example usage or snippet here.

### **Q43. How to chain multiple middleware for one route?**

**Quick answer:** Pass array of functions before handler.

**Example:** `app.get('/path',[auth,logger],fn);`

### **Q44. What is `app.param()`?**

**Quick answer:** Middleware triggered when parameter is present in route.

**Example:** `app.param('id',(req,res,next,id)=>{req.userId=id; next();});`

### **Q45. What is difference between synchronous error and async error handling?**

**Quick answer:** Sync errors bubble automatically. Async errors need `next(err)` or `try/catch`.

**Example:** `try{ await fn(); }catch(e){ next(e); }`

### **Q46. How do you secure sensitive routes?**

**Quick answer:** Use auth middleware with JWT or session validation.

**Example:** `app.get('/dashboard',auth,fn);`

### **Q47. What is difference between res.jsonp and res.json?**

**Quick answer:** jsonp wraps response in callback function for cross-domain requests.

**Example:** `res.jsonp({msg:'ok'});`

### **Q48. What is app.engine()?**

**Quick answer:** Registers template engine.

**Example:** `app.engine('html',require('ejs').renderFile);`

### **Q49. What is express-validator?**

**Quick answer:** Middleware for validating/sanitizing request data.

**Example:** `app.post('/user',[check('email').isEmail()],fn);`

### **Q50. Additional Express.js concept #50**

**Quick answer:** Explanation of an Express.js interview topic.

**Example:** Example usage or snippet here.