

# Express.js Cheat Sheet

Express 5.3.0 (4.22.3 maintenance) · Node.js 18+ minimum, 24/26 LTS · checked 2026-10-29  
Bookshelf examples · ESM, node: prefixes

## Setup & versions

```
npm init -y && npm i express
// package.json: "type": "module"
import express from 'express';
const app = express();
app.get('/', (req, res) => res.send('hi'));
app.listen(3000); // 5.x: callback gets error
```

**App/server split:** app.js exports app, server.js calls listen (tests import the app).  
node --watch --env-file=.env server.js

**Support:** only the latest of each major. v5 ACTIVE since 2025-03-31 (5.1 = latest); v4 MAINTENANCE, latest-4. Node: v5 ≥18, v4 ≥0.10. npm view express dist-tags. No Express 6 announced.

## Routing

```
app.get|post|put|patch|delete(path, ...handlers)
app.all(path, h) app.query(path, h) // HTTP QUERY
app.route('/books').get(list).post(create)
(req,res,next) => next('route') // skip rest of route
```

```
/books/:id      named param, req.params.id
/*splat         wildcard, 1+ segments (array in 5)
/{*splat}       wildcard, 0+ segments
/book{.:ext}     optional part
/~/b\d+$/       RegExp object still fine
5.x: no ? * + () in strings (path-to-regexp 8)
```

```
app.param('id', (req,res,next,id) =>
{...next()})
```

## Routers

```
const r = express.Router({ mergeParams: true,
  caseSensitive: false, strict: false });
r.use(auth); r.get('/', list);
app.use('/api/v1/books', r);
// next('router') leaves this router
```

req.baseUrl = mount path, req.originalUrl = full URL, req.path relative. Version APIs with a router per version; vhost for hosts. Layout: routes → controllers → services.

## Middleware

```
app.use((req, res, next) => { ...; next(); });
app.use('/admin', mw) // path prefix
const limit = (n) => (req,res,next) => {...} // factory
app.get('/x', async (req,res) => { await db() })
```

Order = execution order; end the response or call next(), never both. Types: application, router, error, built-in, third-party. Share data with res.locals. 5.x: rejected promises and thrown errors reach the error handler.

## Body parsing & static

```
app.use(express.json({ limit: '100kb', strict: true }));
app.use(express.urlencoded({ extended: false }));
express.raw() express.text() // type, limit, verify
app.use(express.static('public', {
  maxAge: '1d', index: false, dotfiles: 'ignore',
  extensions: ['html'], fallthrough: true }));
```

5.x: req.body is undefined until a parser runs; defaults extended: false, dotfiles: 'ignore'. Multipart: multer.

## App settings

|                                        |                                        |
|----------------------------------------|----------------------------------------|
| app.set / get / enable / disable       |                                        |
| trust proxy                            | false (default), hops, subnet list, fn |
| query parser                           | 5.x default simple                     |
| etag                                   | weak default; strong, false, fn        |
| x-powered-by                           | disable it                             |
| env                                    | NODE_ENV or development                |
| json spaces / json replacer            | JSON output                            |
| case sensitive routing, strict routing | off                                    |
| views, view engine, view cache         | views                                  |
| subdomain offset                       | 2                                      |
| jsonp callback name                    | deprecated use                         |

## Request & response

**req:** params query body headers cookies signedCookies ip ips hostname protocol secure path baseUrl originalUrl method xhr fresh stale; req.get(h), req.accepts(types), req.is(type), req.range().

**res:** status(201).json(o) send sendStatus set/get type location links vary cookie(n,v,opts) clearCookie redirect(301, url) sendFile(abs,{root}) download(path,name) attachment end render format({...}) append.  
res.format({ json: () => res.json(b), html: () => res.render('b', { b }) });

req.query and req.params are getters in 5.x; res.redirect('back') and res.sendFile are gone.

## Views

```
app.set('views', './views');
app.set('view engine', 'pug'); // pug, ejs, hbs...
app.engine('html', engineFn); // custom engine
res.render('book', { title }, cb);
res.locals.user = req.user; // all views
```

Engines: Pug, EJS, express-handlebars, Nunjucks, @ladjs/consolidate. Escape output; set view cache in production. express generator defaults changed (Jade is now Pug).

## Errors

```
app.use((req, res) => res.status(404).json({
  type: 'about:blank', title: 'Not Found',
  status: 404 }));
app.use((err, req, res, next) => { // 4 args
  if (res.headersSent) return next(err);
  res.status(err.status ?? 500)
    .type('application/problem+json')
    .json({ title: err.message, status:
err.status ?? 500 });
});
```

Register last. Async errors forwarded automatically in 5.x (4.x needs next(err)). Default handler: uses err.status, hides stack unless NODE\_ENV is not production. RFC 9457 problem details.

## APIs & data

|             |                               |
|-------------|-------------------------------|
| GET         | 200 list/item · 404 missing   |
| POST        | 201 + Location                |
| PUT / PATCH | 200 or 204 · replace / merge  |
| DELETE      | 204 · 400/401/403/409/422/429 |

```
const { data, error } =
schema.safeParse(req.body);
if (error) return next(Object.assign(new
Error('bad'), { status: 422 }));
```

Validate with express-validator, Zod, Joi or Ajv; whitelist fields (no mass assignment). Create one pool/client at startup, share it, close it at shutdown; parameterized queries only.

```
await mongoose.connect(process.env.MONGODB_URI);
app.listen(port); // connect before listening
```

Uploads: multer({ storage, limits, fileFilter }).single('cover'); res.download.

## Security

```
app.disable('x-powered-by');
app.use(helmet());
app.use(rateLimit({ windowMs: 15*60e3, limit: 100 }));
app.use(cors({ origin: ['https://app.example.com'],
  credentials: true, maxAge: 600 }));
app.use(session({ secret:
process.env.SESSION_SECRET,
  resave: false, saveUninitialized: false,
  store,
  cookie: { httpOnly: true, secure: true,
sameSite: 'lax' } }));
```

CSRF: csrf is deprecated; use csrf-csrf / csrf-sync or SameSite. TLS at the proxy + trust proxy so secure cookies work. Never origin: \* with credentials. Keep deps patched (npm audit); security releases are monthly.

## Authentication

**Session** Passport local + express-session + store (connect-redis, connect-mongo); serializeUser/deserializeUser; req.logout(cb); regenerate session at login.

**JWT** pin algorithms, short expiry, refresh rotation, verify iss/aud; jose or express-jwt; Authorization: Bearer.

**Passwords** argon2 or bcrypt (async); never log or return hashes; uniform login errors.

## Testing

```
import { test } from 'node:test';
import request from 'supertest';
import { app } from './app.js';
test('GET /api/v1/books', async () => {
  const res = await
request(app).get('/api/v1/books')
  .expect(200).expect('Content-Type', /json/);
});
```

Import the app, never listen. node --test. In-memory Mongo / Testcontainers; Playwright for E2E.

## Production

```
NODE_ENV=production # caches views, terse errors
app.use(compression()); // or at the proxy
app.get('/healthz', (req, res) =>
res.sendStatus(200));
process.on('SIGTERM', () => server.close() =>
process.exit(0));
// FROM node:lts-alpine; npm ci --omit=dev; USER node
// CMD ["node","server.js"]
```

Liveness vs readiness; stop accepting, drain, close pools, then exit. Proxy: app.set('trust proxy', 1). Cluster / PM2 or one process per container. Avoid sync calls; set timeouts; cache.

## Debugging & logging

```
DEBUG=express:* ,router,router:* node server.js
app.use(morgan('combined')); // access log
app.use(pinoHttp()); // structured JSON
```

Use debug('bookshelf:db') in your own code. Log to stdout; do not log secrets or full bodies.

## Changed since older tutorials

|         |                                                                                                                                                              |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| removed | app.del → delete · res.json(status,o) → status().json() · req.param() · res.sendFile → sendFile · app.router · createServer · app.configure                  |
| paths   | /* → /*splat, /:id? → /{:id}; no inline regex                                                                                                                |
| async   | no wrapper needed for async handlers                                                                                                                         |
| bodies  | body-parser package → express.json(); req.body may be undefined; extended: false                                                                             |
| static  | dotfiles: 'ignore'                                                                                                                                           |
| tooling | generator uses Pug not Jade; csrf deprecated; Mongoose callbacks removed; Heroku free tier / mLab gone; codemods: npx codemod @expressjs/v5-migration-recipe |