

Express.js Cheat Sheet

worksheetdownload.com

Routing · Middleware · REST · Static Files



SETUP & APP

```
npm install express
import express from 'express'
const app = express()
app.use(express.json())
app.use(express.urlencoded({ extended:
true })))
app.listen(3000, () =>
console.log('Running on 3000'))
```

ROUTING

```
app.get('/users', getUsers)
app.post('/users', createUser)
app.put('/users/:id', updateUser)
app.delete('/users/:id', deleteUser)
app.all('/admin/*', adminMiddleware,
adminController)
app.route('/items').get(list).post(create)
```

MIDDLEWARE

```
app.use((req, res, next) => {
console.log(req.method, req.url)
next()          // call next middleware
})
app.use('/api', apiRouter)
app.use('/static',
express.static('public'))
Order matters  middleware runs
top-to-bottom
```

REQUEST OBJECT

```
req.params.id           // route :id
req.query.page           // ?page=2
req.body.email           // parsed
JSON/form
req.headers.authorization
req.cookies.session     //
(cookie-parser)
req.ip req.method req.path
req.hostname
```

RESPONSE OBJECT

```
res.json({ users: [] })           // 200
JSON
res.status(201).json({ id: 1 })
res.status(404).json({ error: 'Not
found' })
res.send('Hello')                // text
res.sendFile(path)               // file
res.redirect('/login')
res.set('X-Header', 'value')
```

ROUTER

```
// users.js
const router = express.Router()
router.get('/', listUsers)
router.post('/', createUser)
router.get('/:id', getUser)
export default router
// app.js
app.use('/api/users', usersRouter)
```

ERROR HANDLING

```
app.use((err, req, res, next) => {
console.error(err.stack)
res.status(err.status ?? 500).json({
error: err.message
})
})
4-arg middleware = error handler;
must be last
next(err) to pass error to handler
```

ROUTE PARAMS & QUERY

```
app.get('/users/:id/posts/:postId',
(req, res) => {
const { id, postId } = req.params
const { page = 1, limit = 10 } =
req.query
res.json({ userId: id, postId, page,
limit })
})
app.param('id', (req, res, next, id) =>
{
req.user = User.findById(id)
next()
})
```

STATIC FILES

```
app.use(express.static('public'))
app.use('/assets',
express.static('dist/assets'))
// Serve with options:
app.use(express.static('public', {
maxAge: '1d',
etag: true
}))
Files in 'public' served at root URL
```

BODY PARSING

```
app.use(express.json({ limit: '10mb' })))
app.use(express.urlencoded({ extended:
true })))
// Multipart / file upload:
import multer from 'multer'
const upload = multer({ dest: 'uploads/'
})
app.post('/upload',
upload.single('file'), handler)
```

COMMON MIDDLEWARE

```
helmet()           // security headers
cors()             // CORS headers
morgan('dev')      // request logging
compression()      // gzip responses
cookie-parser      // parse cookies
express-rate-limit // rate limiting
express-validator  // input validation
```

COMMON GOTCHAS

Error handler must have 4 args
(err, req, res, next)
Missing next() call hangs request
indefinitely
Order of app.use() matters put
catch-all last
express.json() body is undefined
without middleware
async errors need try/catch +
next(err) pattern