

Express.js Quiz Cheatsheet (With Examples)

1. Express App Setup

```
-----  
const express = require('express');  
const app = express();  
app.use(express.json());  
app.listen(3000, () => console.log('Server running'));
```

2. Routing Basics

```
-----  
app.get('/', (req, res) => res.send("Home"));  
app.post('/user', (req, res) => res.json(req.body));
```

3. req Object

```
-----  
- req.params: URL params (e.g. /user/:id) req.params.id  
- req.query: Query string (e.g. ?q=abc) req.query.q  
- req.body: JSON body data req.body.name  
- req.method: HTTP verb used  
- req.url: Full request URL
```

4. res Object

```
-----  
- res.send(): Send plain response  
- res.json(): Send JSON  
- res.status(): Set HTTP code  
- res.redirect(): Redirect user  
- res.end(): End response
```

5. Common HTTP Status Codes

```
-----  
200 - OK: Success (GET/PUT)  
201 - Created: Success (POST)  
204 - No Content: Success, no data to return  
400 - Bad Request: Invalid input  
401 - Unauthorized: Login required  
403 - Forbidden: No permission  
404 - Not Found: Item or route missing  
500 - Server Error: App crashed
```

6. Middleware with app.use()

```
-----  
// Logger middleware  
app.use((req, res, next) => {
```

```
console.log(req.method, req.url);
next();
});

// Auth middleware for specific route
const auth = (req, res, next) => {
  if (req.query.admin === 'true') next();
  else res.status(403).send("Forbidden");
};

app.get('/admin', auth, (req, res) => {
  res.send("Welcome Admin");
});
```

7. Folder Structure

```
-----
project/
  controllers/
  routes/
  models/
  db/
  index.js
```

8. Route with params, query, and body

```
-----
app.patch('/product/:id', (req, res) => {
  const id = req.params.id;
  const search = req.query.search;
  const { name } = req.body;
  res.json({ id, search, name });
});
```