

Express.js Routes Guide

1. GET Route

Definition: Retrieves data from the server.

Use: Show a list of users, fetch a single product, etc.

Example:

```
app.get('/users', (req, res) => {  
  res.send('User list');  
});
```

2. POST Route

Definition: Submits new data to the server.

Use: Add a new user, submit a form, etc.

Example:

```
app.post('/users', (req, res) => {  
  res.send('User created');  
});
```

3. PUT Route

Definition: Updates existing data.

Use: Update a user's info, modify a post, etc.

Example:

```
app.put('/users/:id', (req, res) => {  
  res.send(`User ${req.params.id} updated`);  
});
```

4. DELETE Route

Definition: Deletes data.

Use: Remove a user, delete a product, etc.

Example:

```
app.delete('/users/:id', (req, res) => {  
  res.send(`User ${req.params.id} deleted`);  
});
```

Express.js Routes Guide

```
});
```

5. Route Parameters

Definition: Dynamic parts of a route (e.g. `:id`, `:username`).

Use: Get user by ID, product by slug, etc.

Example:

```
app.get('/product/:id', (req, res) => {  
  res.send(`Product ID: ${req.params.id}`);  
});
```

6. Query Parameters

Definition: Passed in URL after `?`, used for filters, search, etc.

Use: `/search?q=laptop&page=2`

Example:

```
app.get('/search', (req, res) => {  
  res.send(`Search: ${req.query.q}, Page: ${req.query.page}`);  
});
```

7. Middleware in Route

Definition: Functions that run before route handlers.

Use: Authentication, logging, validation, etc.

Example:

```
const logger = (req, res, next) => {  
  console.log(`${req.method} ${req.url}`);  
  next();  
};  
  
app.get('/dashboard', logger, (req, res) => {  
  res.send('Dashboard');  
});
```

Express.js Routes Guide

8. Express Router (Modular Routes)

Definition: A mini Express app for organizing routes in separate files.

Use: Large projects with separate route files per feature.

Example:

```
// routes/userRoutes.js

const express = require('express');
const router = express.Router();

router.get('/', (req, res) => res.send('All users'));
router.post('/', (req, res) => res.send('Create user'));

module.exports = router;
```

```
// app.js

const userRoutes = require('./routes/userRoutes');
app.use('/users', userRoutes);
```

9. Wildcard Route (404 Not Found)

Definition: Catches all undefined routes.

Use: Show a custom "Not Found" page.

Example:

```
app.use((req, res) => {
  res.status(404).send('404 - Page Not Found');
});
```

10. Multiple Handlers for One Route

Definition: Chain multiple middleware functions for one route.

Use: Clean up logic, authentication + data processing.

Example:

```
app.get('/profile', authMiddleware, logMiddleware, (req, res) => {
```

Express.js Routes Guide

```
res.send('User profile');  
});
```

11. All Methods for a Path (`app.route`)

Definition: Chain methods for a single route path using `app.route()`.

Use: Cleaner syntax for same-path routes.

Example:

```
app.route('/article')  
  .get((req, res) => res.send('Get article'))  
  .post((req, res) => res.send('Create article'))  
  .put((req, res) => res.send('Update article'));
```