**cookie-parser Middleware in Express.js**

cookie-parser is a third-party middleware in Express.js that **parses cookies** attached to incoming client requests.

**1. Installing cookie-parser**

Before using it, install the package:

npm install cookie-parser

**2. Basic Usage**

Import and use cookie-parser in your Express app.

const express = require('express');

const cookieParser = require('cookie-parser');

const app = express();

// Use cookie-parser middleware

app.use(cookieParser());

app.get('/', (req, res) => {

res.send('Welcome! Use /set-cookie and /get-cookie routes.');

});

// Set a cookie

app.get('/set-cookie', (req, res) => {

res.cookie('username', 'JohnDoe'); // Set a cookie

res.send('Cookie has been set!');

});

// Get a cookie

app.get('/get-cookie', (req, res) => {

res.json(req.cookies); // Retrieve cookies

});

// Clear a cookie

app.get('/clear-cookie', (req, res) => {

res.clearCookie('username'); // Delete the 'username' cookie

res.send('Cookie has been cleared!');

});

app.listen(3000, () => console.log('Server running on port 3000'));

**3. How It Works**

**✅ Setting a Cookie (/set-cookie)**

* Sends a cookie to the client.
* The browser stores it and sends it with future requests.

**✅ Retrieving Cookies (/get-cookie)**

* Extracts cookies from req.cookies.
* If cookie-parser is not used, req.cookies will be undefined.

**✅ Clearing Cookies (/clear-cookie)**

* Deletes the username cookie from the client.

**4. Setting Cookies with Options**

You can pass options when setting cookies:

app.get('/set-cookie-options', (req, res) => {

res.cookie('sessionID', 'abc123', {

maxAge: 60 \* 1000, // Expires in 1 minute

httpOnly: true, // Prevents client-side access

secure: false, // Set to true in HTTPS

sameSite: 'Strict' // Restrict cross-site cookies

});

res.send('Cookie with options has been set!');

});

📌 **Options Explained:**

| **Option** | **Description** |
| --- | --- |
| maxAge | Cookie expiration time in milliseconds |
| httpOnly | Prevents JavaScript from accessing the cookie |
| secure | Sends the cookie only over HTTPS |
| sameSite | Prevents cross-site cookie sharing |

**5. Signed Cookies (Secure Cookies)**

To protect cookies from tampering, use **signed cookies**.

**✅ Enable Signed Cookies**

app.use(cookieParser('mySecretKey')); // Provide a secret key

**✅ Set a Signed Cookie**

app.get('/set-signed-cookie', (req, res) => {

res.cookie('secureData', '12345', { signed: true });

res.send('Signed cookie has been set!');

});

**✅ Retrieve Signed Cookies**

app.get('/get-signed-cookie', (req, res) => {

res.json(req.signedCookies); // Use req.signedCookies

});

📌 **Unsigned cookies appear in req.cookies, but signed cookies appear in req.signedCookies.**

**6. Summary**

| **Feature** | **Example** |
| --- | --- |
| **Install cookie-parser** | npm install cookie-parser |
| **Set a basic cookie** | res.cookie('username', 'JohnDoe') |
| **Retrieve cookies** | req.cookies |
| **Delete cookies** | res.clearCookie('username') |
| **Set cookie expiration** | { maxAge: 60000 } |
| **Secure HTTP-only cookie** | { httpOnly: true, secure: true } |
| **Signed cookies (tamper-proof)** | res.cookie('key', 'value', { signed: true }) |

Would you like any additional examples? 🚀