

Experiment -8: Set and delete cookie using Express.js

- 1. Aim:** To set, get and delete the cookie using Express.js
- 2. Objective:** To teach students to use Express.js for setting and deleting the cookie.
- 3. Lab outcome:** Students will be able to create back-end applications using Node.js/ Express (PO3, PO5, PSO3, PSO4)
- 4. Prerequisite:** Node.js, Express.js
- 5. Requirements:** The following are the requirements –
 - PC/Laptop
 - Visual Studio Code
 - Browser

6. Pre-Experiment Theory:

Cookies are simple, small files/data that are sent to client with a server request and stored on the client side. Every time the user loads the website back, this cookie is sent with the request. This helps us keep track of the user's actions.

To use cookies with Express, we need the cookie-parser middleware. To install it, use the following code –

npm install --save cookie-parser

Now to use cookies with Express, we will require the **cookie-parser**. cookie-parser is a middleware which parses cookies attached to the client request object. To use it, we will require it in our **index.js** file; this can be used the same way as we use other middleware.

cookie-parser parses Cookie header and populates **req.cookies** with an object keyed by the cookie names. To set a new cookie, let us define a new route in Express app like –

```
var express = require('express');
var app = express();
app.get('/',
function(req, res){
  res.cookie('name', 'express').send('cookie set'); //Sets name = express
});
```

```
app.listen(3000);
```

- **Deleting Existing Cookies**

```
var express = require('express');
```

```
var app = express();
app.get('/clear_cookie_foo',
function(req, res){
  res.clearCookie('foo');
  res.send('cookie foo cleared');
});
app.listen(3000);
```

7. Laboratory Exercise:

A. Procedure

- Check version of node.js (node -v)
- Check version of npm (npm -v)
- Create package.json file (npm init)
- Install express from terminal using following commands -
 - npm install express –save
- Install nodemon – the command line tool using the following command:
 - npm install nodemon
- Open vs-code and open the folder.
- Create a .js file.
- Write the code.
- To get the output go to the terminal and execute the command=>
 - nodemon filename.js
- Visit the browser to see the output.

B. Program Code

Write a code to set, get and delete the cookie in express.

8. Post Experimental Exercise-

- What is cookie-parser in express? Why it is used?
- What is the cookie value?

9. Results/Observations/Program output:

Present the program code and output

10. Conclusion:

Write what was performed in the experiment.

11. References:

- <https://nodejs.org/en/>
- <https://www.youtube.com/watch?v=ZySsdm576wE>
- <https://www.youtube.com/watch?v=5uZDLL0zaWg>
- https://www.w3schools.com/nodejs/nodejs_filesystem.asp
- <https://codeforgeek.com/node-js-tutorial-step-by-step/>

7B.Program Code

Creating a web server using express and displaying HelloWorld in a web page:

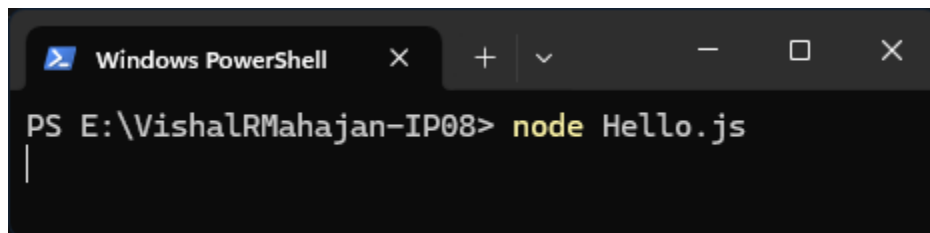
hello.js

```
const express = require('express')

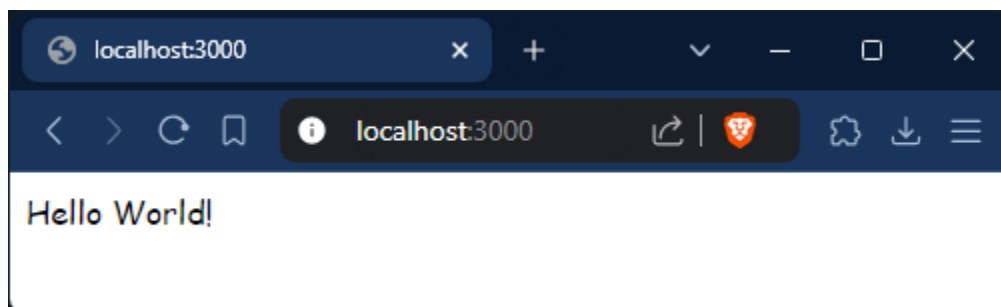
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Hello World!');
}).listen(port)
```

Running the server



Output on Port 3000



1. Write a code to set, get and delete the cookie in express.

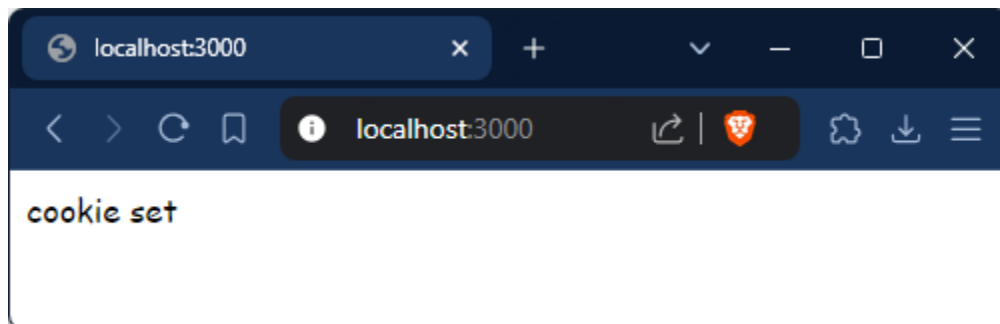
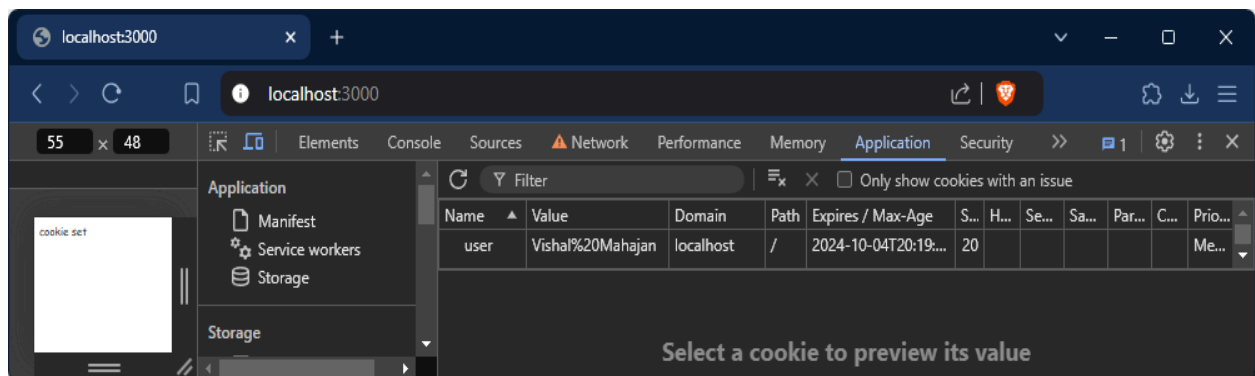
SetCookie.js with Browser and Application Output

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

const app = express();
const port = 3000;

app.use(cookieParser());

app.get('/', (req, res) => {
  res.cookie('user', 'Vishal Mahajan', {maxAge: 24*60*60*1000}).send('cookie set');
}).listen(port)
```

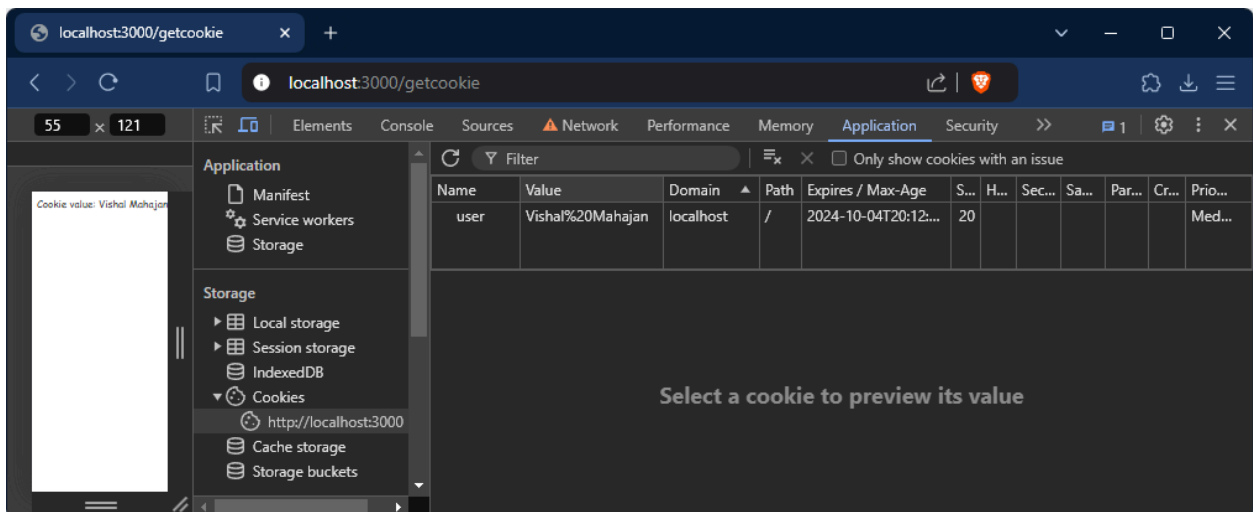
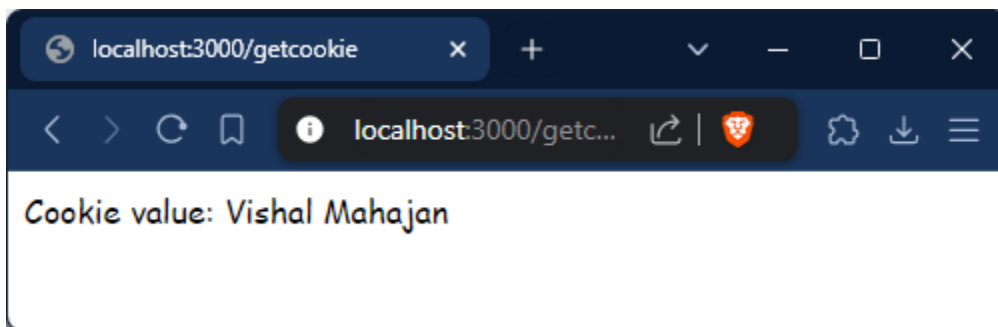


GetCookie.js with Browser and Application Output

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

const app = express();
const port = 3000;
app.use(cookieParser());

app.get('/getcookie', (req, res) => {
  const userCookie = req.cookies.user;
  if (userCookie) {
    res.send(`Cookie value: ${userCookie}`);
  }
  else {
    res.send('Cookie not set');
  }
}).listen(port)
```



DeleteCookie.js with Browser and Application Output

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

const app = express();
const port = 3000;

app.use(cookieParser());

app.get('/deleteCookie', (req, res) => {
  res.clearCookie('user').send('Cookie deleted');
}).listen(port)
```

