

## Week-8

**Aim:** Maintaining the transactional history of any user is very important. Explore the various session tracking mechanism (Cookies, HTTP Session)

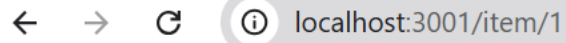
### Codes:

#### cookieApp.js

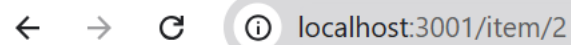
```
const express = require('express');
const cookieParser =
require('cookie-parser'); const
app = express();
const PORT =
3001;
app.use(cookie
eParser());
// Simulate visiting an
item app.get('/item/:id',
(req, res) => {
let history = req.cookies.history ? JSON.parse(req.cookies.history) : [];
// Add the visited item
history.push(`Item
${req.params.id}`);
// Log the history before updating
cookie console.log('Cookie
history:', history);
// Limit to last 5 items
if (history.length > 5) history.shift();
// Update cookie
res.cookie('history', JSON.stringify(history), { httpOnly:
true }); res.send(`Visited Item ${req.params.id}`);
});
// Show browsing
history
app.get('/history', (req,
res) => {
const history = req.cookies.history ?
JSON.parse(req.cookies.history) : []; res.send(`Browsing
History (Cookies): ${history.join(', ')}`);
```

```
});  
app.listen(PORT, () => {  
  console.log(` Cookie-based app running at http://localhost:${PORT}`);  
});
```

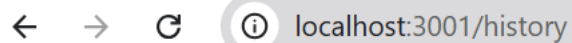
## **OUTPUT :**



Visited Item 1



Visited Item 2



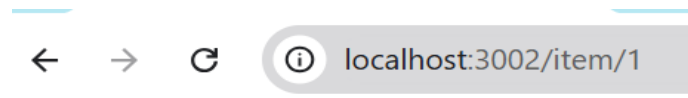
Browsing History (Cookies): Item 1, Item 2

## sessionApp.js

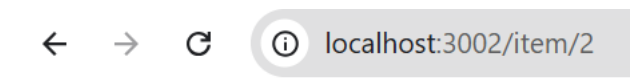
```
const express = require('express');
const session =
require('express-session');
const app = express();
const PORT = 3002;
// Session
middleware
setup
app.use(sessi
on({
secret:
'sessionSe
cretKey',
resave:
false,
saveUniniti
alized:
false,
cookie: { maxAge: 600000 } // 10 minutes
}));
// Simulate visiting an
item app.get('/item/:id',
(req, res) => {
if (!req.session.history) req.session.history = [];
// Add the visited item
req.session.history.push(`Item
${req.params.id}`);
// Log session history to the console for debugging
console.log(req.session.history);
// Limit to last 5 items
if (req.session.history.length > 5)
req.session.history.shift(); res.send(`Visited Item
${req.params.id}`);
});
// Show session-based
history
app.get('/history', (req,
res) => {
const history = req.session.history || [];
res.send(`Session Browsing History: ${history.join(', ')}');
});
// Clear session
app.get('/logout',
(req, res) => {
req.session.dest
```

```
roy(err => {  
  if (err) return res.send('Logout  
  failed'); res.send('Session  
  cleared');  
});  
});  
  
app.listen(PORT, () => {  
  console.log(` Session-based app running at http://localhost:${PORT}`);  
});
```

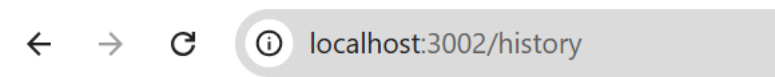
### **OUTPUT :**



Visited Item 1



Visited Item 2



Session Browsing History: Item 1, Item 2, Item 1, Item 2