

# Practical – 13

GitHub Link: <https://github.com/Devsharma511/NodeJs.git>

## Task 1: Streaming & Backpressure + Client Abort-Aware Middleware

Goal: Implement a streaming endpoint producing NDJSON. If a client disconnects, the middleware must stop work.

## Task 2: Content Negotiation Middleware (JSON vs. XML)

Goal: Return JSON or XML based on Accept header with a reusable negotiate middleware.

## Project Structure:

app.js:

```
JS app.js U X
Practical-13 > JS app.js > app.use() callback > res.on('finish') callback
1 import express from 'express';
2
3 const app = express();
4 const PORT = 3000;
5
6 app.use((req, res, next) => {
7   req.id = req.headers['x-request-id'] || Math.random().toString(36).slice(2);
8   res.setHeader('X-Request-Id', req.id);
9   const start = process.hrtime.bigint();
10  res.on('finish', () => {
11    const end = process.hrtime.bigint();
12    const ms = Number(end - start) / 1e6;
13    if (!res.headersSent) {
14      res.setHeader('X-Response-Time-ms', ms.toFixed(2));
15    }
16  });
17  next();
18 });
19
20 // Task 1: Streaming & backpressure + client abort awareness
21 app.get('/stream', (req, res) => {
22   res.setHeader('Content-Type', 'application/x-ndjson');
23   let count = 0;
24   const interval = setInterval(() => {
25     if (res.writableEnded) {
26       clearInterval(interval);
27       return;
28     }
29     const data = JSON.stringify({ number: count });
30     const flushed = res.write(data + '\n');
31     count++;
32     if (count > 50) {
33       clearInterval(interval);
34       res.end();
35     }
36     if (!flushed) {
37       clearInterval(interval);
38     }
39   }, 100);
40   req.on('close', () => {
41     clearInterval(interval);
42   });
43 });
44
45 JS app.js U X
Practical-13 > JS app.js > app.use() callback > res.on('finish') callback
45 // Task 2: Reusable content negotiation middleware
46 function negotiate(req, res, next) {
47   res.negotiate = (obj) => {
48     const accept = req.headers['accept'] || '';
49     if (accept.includes('application/xml')) {
50       res.type('application/xml');
51       const xml = `<root>\n${Object.entries(obj)
52         .map(([key, value]) => `<${key}>${value}</${key}>`)
53         .join('\n')}\n</root>`;
54       res.send(xml);
55     } else {
56       res.type('application/json');
57       res.send(obj);
58     }
59   };
60   next();
61 }
62
63 app.use('/content', negotiate);
64
65 app.get('/content', (req, res) => {
66   res.negotiate({ message: 'Hello, Negotiated World!' });
67 });
68
69 app.use((err, req, res, next) => {
70   if (!res.headersSent) {
71     res.status(500).json({ error: err.message || 'Server Error' });
72   }
73 });
74
75 app.listen(PORT, () => {
76   console.log(`Server running on port ${PORT}`);
77 });
78
```

## Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

● PS C:\Users\sharm\nodejs> cd Practical-13
○ PS C:\Users\sharm\nodejs\Practical-13> node app
Server running on port 3000
█
```

## Task 1:

```
stream x
C: > Users > sharm > Downloads > stream
1  {"number":0}
2  {"number":1}
3  {"number":2}
4  {"number":3}
5  {"number":4}
6  {"number":5}
7  {"number":6}
8  {"number":7}
9  {"number":8}
10 {"number":9}
11 {"number":10}
12 {"number":11}
13 {"number":12}
14 {"number":13}
15 {"number":14}
16 {"number":15}
17 {"number":16}
18 {"number":17}
19 {"number":18}
20 {"number":19}
21 {"number":20}
22 {"number":21}
23 {"number":22}
24 {"number":23}
25 {"number":24}
26 {"number":25}
27 {"number":26}
28 {"number":27}
29 {"number":28}
30 {"number":29}
31 {"number":30}
```

## Task 2:

```
localhost:3000/content x +
localhost:3000/content

This XML file does not appear to have any style information associated with it. The document tree is shown below.
▼<root>
  <message>Hello, Negotiated World!</message>
</root>
```