WEEK 6

● Create gateway using node js

● Create cron jobs using Node js

● Blocking vs Non Blocking methods

● Webpack

LAB Exercise:

1) Initialize a new node js project and install express , http-proxy-middleware node modules.

```
>mkdir Project2
>cd Project2
>npm init -y
>npm install express http-proxy-middleware
```

2) Create the API gateway for a backend service

```
//gateway.js
const express = require('express');

const { createProxyMiddleware } = require('http-proxy-middleware');

const app = express();
const PORT = 3000; // Port for your API Gateway
// Example route for a different backend service
app.use('/service1', createProxyMiddleware({ target: 'http://localhost:3001', changeOrigin: true }));
// Basic route for the gateway itself
app.get('/', (req, res) => {
res.send('API Gateway is running!');
});
app.listen(PORT, () => {
console.log(`API Gateway listening on port ${PORT}`);
```

});

3) Create a backend service to fulfill the gateway request

// service1.js

const express = require('express');

const app = express();

const PORT = 3001;

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

res.send('Hello from Service 1!');

});

app.listen(PORT, () => {

console.log(`Service 1 listening on port ${PORT}`);

});

4) Run the gateway and access the service

start the service

>node service1.js

start the gateway in separate terminal

>node gateway.js

access the service

http://localhost:3000/service1

5) Create cron job using node js

Install the node-cron package

> npm install node-cron

Import the node-cron module. Define the function to be executed.

Schedule the cron job.

//cron.js

const cron = require('node-cron');

const myScheduledTask = () => {

console.log('Cron job executed at:', new Date().toLocaleString());

```
// Add your task logic here, e.g., send emails, update data, etc.
};
cron.schedule('* * * * *', myScheduledTask); //run every minute
//cron.schedule('0 * * * *', myScheduledTask); //run every hour
```

6) Blocking vs Non-Blocking methods

```
//read using callback (Non-Blocking)
fs.readFile('data.txt', 'utf-8', function (err, data)
{
if(err){
console.log('some error');
}
else
{
console.log(data);
}
});
//read using try -catch block (Blocking)
try{
let data = fs.readFileSync('data.txt', 'utf8');
console.log(data);
}
catch(err)
{
console.log(err);
}
```

7) Webpack

Install webpack and include it

```
>npm install --save-dev webpack
>npm install webpack-cli
```

Create a new file named webpack.config.js

```
// webpack.config.js
const path = require('path');
module.exports = {
target: 'node',
entry: './index.js',
output: { filename: 'Project2_bundle.js',
path: path.resolve(__dirname, 'dist'),
},
// Additional configuration goes here
};
```

Update the build command in the package.json file to run the Webpack configuration:

```
"scripts": { "build": "webpack --config webpack.config.js" }
```

execute the npm run build command to build the bundle.

```
> npm run build
```

the project bundle