

## Lesson 06 Demo 04

### Working with Child API

**Objective:** To create cluster API for understanding the multiprocessing function

**Tools required:** Node Package Manager and Visual Studio Code

**Prerequisites:** Basic Linux Commands, NPM commands, JavaScript, HTTP module, child process module, and multiprocessing

Steps to be followed:

1. Create cluster API for multi-core servers

#### Step 1: Create cluster API for multi-core servers

- 1.1 Use the code below to create cluster API without a child process:

```
function longComputation() {  
  let multiplication = 1;  
  for (let num = 1; num <= 1e10; num++) {  
    multiplication *= num  
  }  
  
  return multiplication;  
}
```

```
console.log((new Date()).toString());  
console.log(">>> ", longComputation());  
console.log((new Date()).toString());
```

- 1.2 Execute the code to get the following result:

```
databaseProject % node forking.js  
10:39:01 GMT+0530 (India Standard Time)  
>>> Infinity  
10:39:16 GMT+0530 (India Standard Time)
```

It took 16 seconds to compute and block the main thread.

1.3 Use the code below to make cluster API with the child process fork method:

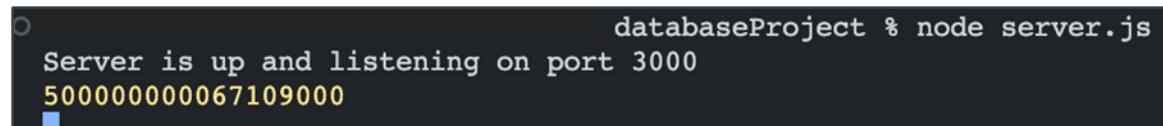
```
const http = require('http');
const { fork } = require('child_process');
const SERVER_PORT = 3000;
const SERVER_HOSTNAME = "127.0.0.1";

const server = http.createServer();
server.on("request", (req, res) => {
  if (req.url === "/compute") {
    let startDate = new Date();
    const compute = fork("./compute.js");
    compute.send("start");
    compute.on("message", (value) => {

      let endTime = new Date();
      res.setHeader('Content-Type', 'application/json')
      res.end(JSON.stringify({
        "startTime": startDate.toTimeString(),
        "endTime": endTime.toTimeString(),
        "value": `Sum from 1 to ${1e9}: ${value}`
      }))
    })
  }
})

server.listen(SERVER_PORT, SERVER_HOSTNAME, () => {
  console.log(`Server is up and listening on port ${SERVER_PORT}`);
})
```

1.4 Execute the code to get the following result:



```
databaseProject % node server.js
Server is up and listening on port 3000
500000000067109000
```

By following these steps, you have successfully created cluster API for understanding the multiprocessing function.