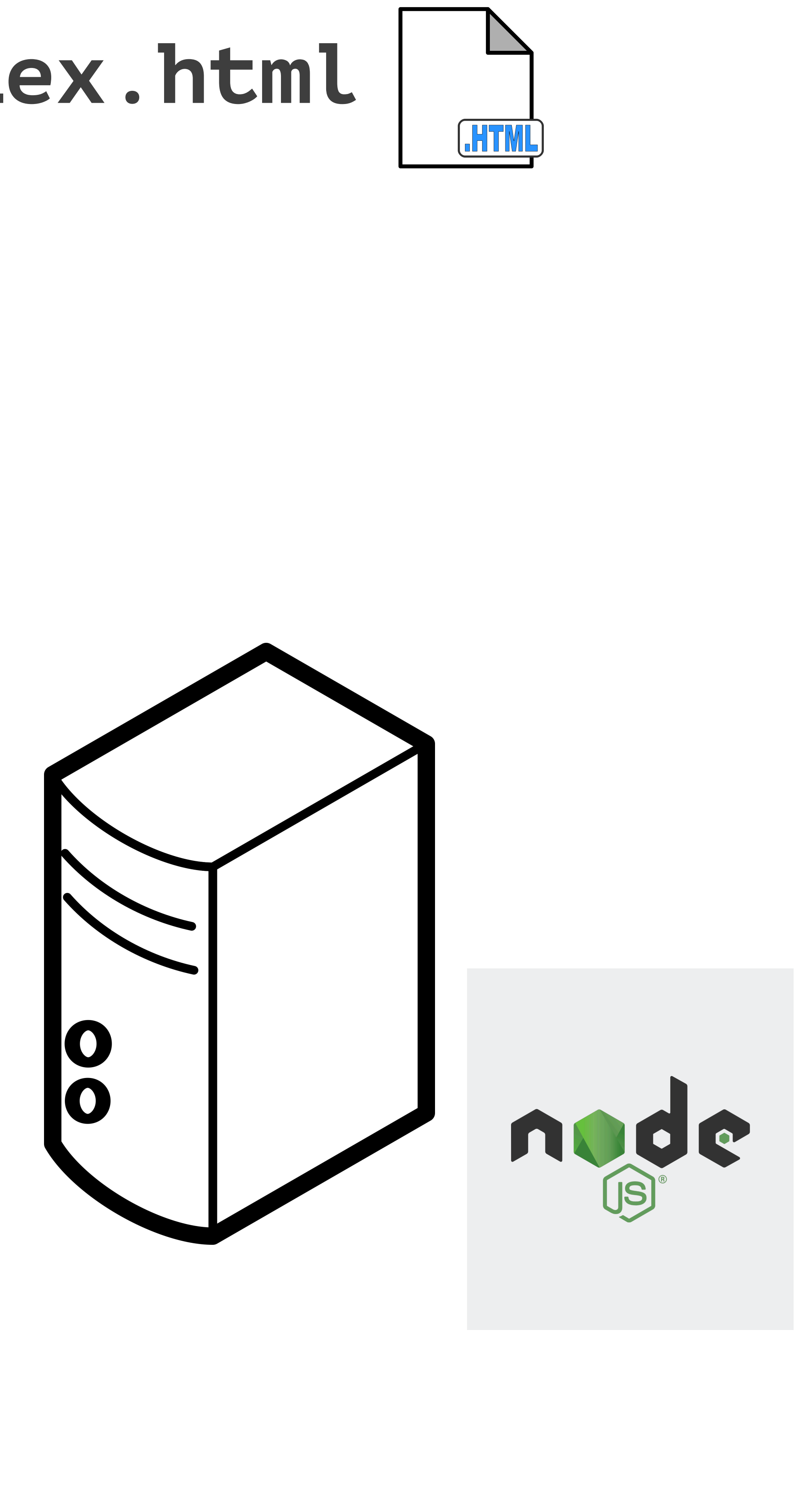
EXPRESS.JS

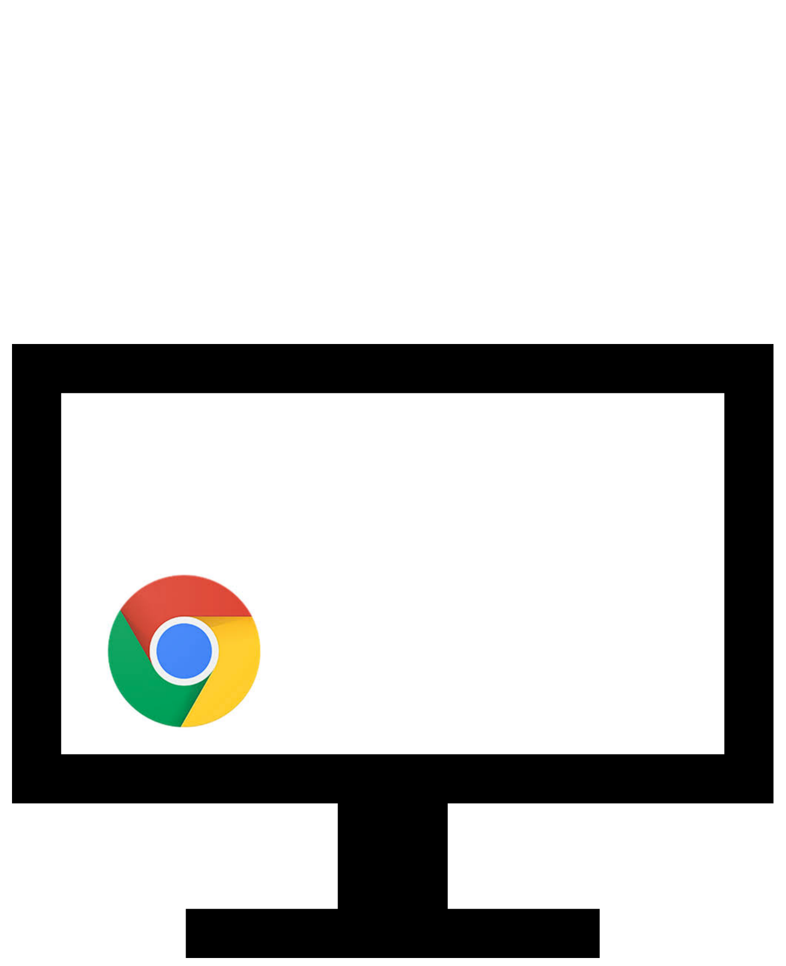
`index.html`

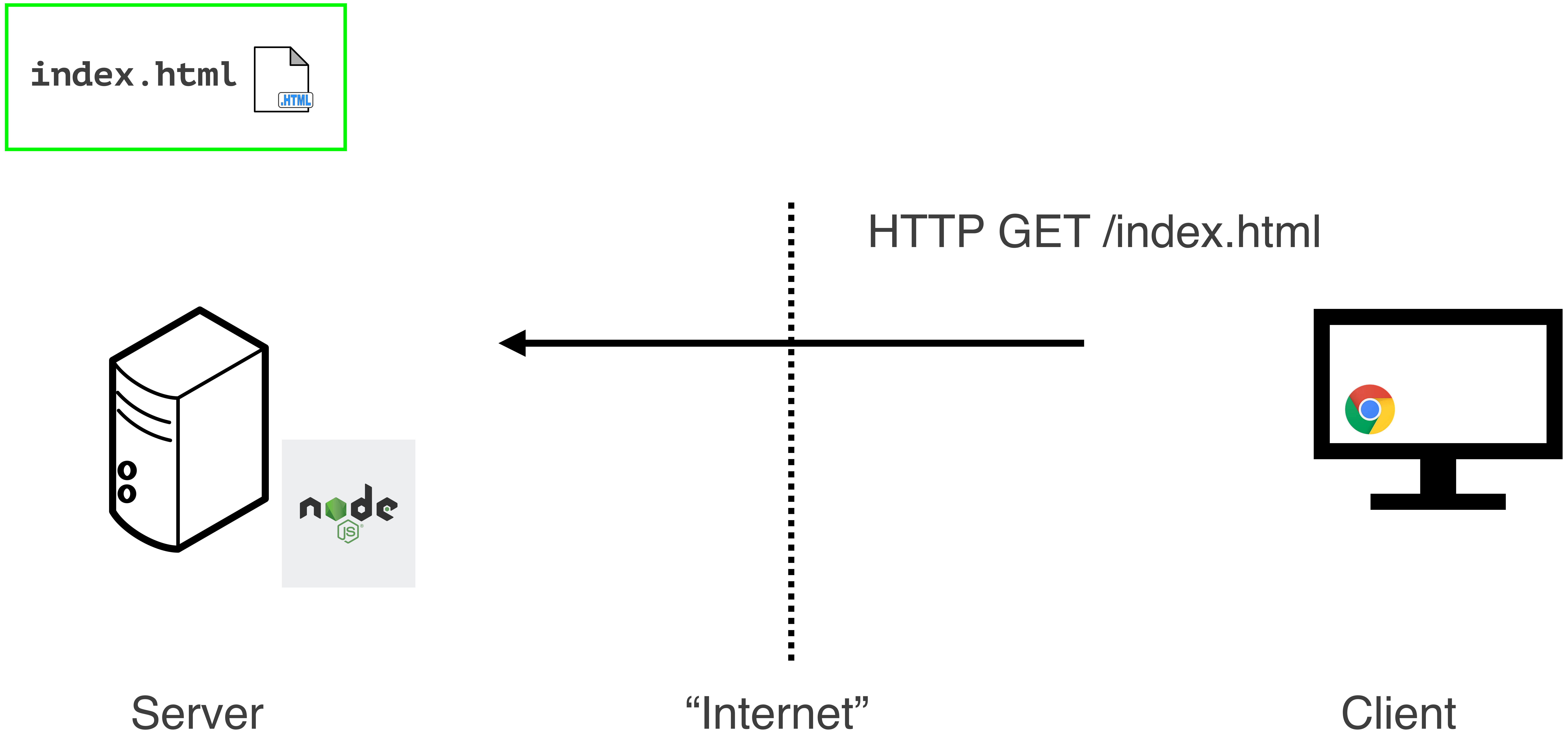


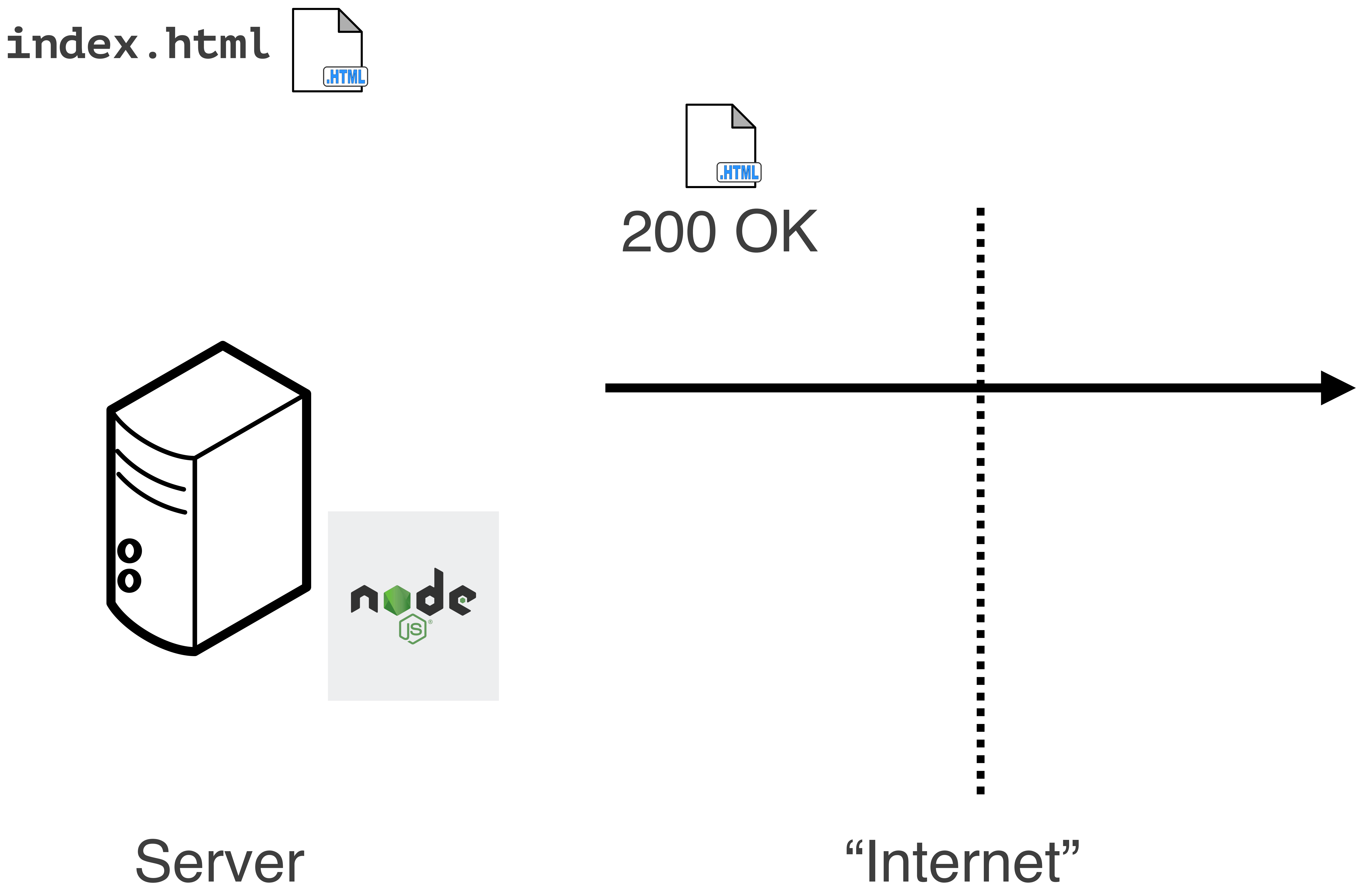
Server

“Internet”

Client




index.html
.HTML
HTTP GET /index.html
node JS
Server
“Internet”
Client


index.html
.HTML
.HTML
200 OK
Server
"Internet"
Client

HTTP REQUEST

just a message with a certain format

verb URI

headers

```
POST /docs/1/related HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

bookId=12345&author=Nimit
```

body

(from http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html)

EXPRESS

- “A Node.js web application framework”
  - Meaning: a library you install and use in Node, helps you build “web apps”
- Our Goal: Use Node to build a server that will…
  - Handle incoming requests (ie: listen, act)
  - Send responses (back to client)
  - … among other things (later!)
- Express helps us accomplish that goal

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

const app = express()

app.listen(3000)
```

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

const app = express()

app.get('/', (req, res, next) => {
  res.send(`<h1>Welcome to the main page</h1>`)
})

app.listen(3000)
```

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

const app = express()

app.get('/', (req, res, next) => {
  res.send(`<h1>Welcome to the main page</h1>`)
})

app.get('/puppies', (req, res, next) => {
  res.send(`<h1>Welcome to the puppies page</h1>`)
})

app.listen(3000)
```

DEMO

SUMMARY

- We can use Node to easily create servers
- The `express` library makes it even easier to write (and subsequently maintain) our server code