

Backend-Assignment Week 2

Name : Prak Pychey

Group : SE G3

Lecturer : Kim Ang Kheang

Exercise 1

Q1 – Here's the error it was shown in terminal when I tried to access localhost:3000 :

```
[pychey@Prak-Pychey EX-1 % node server.js
Server running on http://localhost:3000
/Users/pychey/Downloads/Backend-W2/EX-1/server.js:6
  return res.endd();
              ^
TypeError: res.endd is not a function
    at Server.<anonymous> (/Users/pychey/Downloads/Backend-W2/EX-1/server.js:6:16)
    at Server.emit (node:events:524:28)
    at parserOnIncoming (node:_http_server:1153:12)
    at HTTPParser.parserOnHeadersComplete (node:_http_common:117:17)

Node.js v22.12.0
pychey@Prak-Pychey EX-1 %
```

The 6th line of code that caused it, which was `res.endd();`.

Q2 – Purpose of `res.write` is it writes response body out to client without signaling that the response is finished yet, meaning it's used to write response body to client multiple times. While `res.end` signal the response is finished and ready to be sent, close connection if necessary.

Q3 – When `res.end` is not called at all, client wouldn't know that the response is finished and wait until a timeout occurs.

Q4 – We use `http.createServer` instead of calling function directly because `http.createServer` handle network infrastructure needed to accept connections, handle http requests and complex http protocol and more. While calling function directly won't handle these things like network management, http requests.

Q5 – For server to be made more resilient to such errors during development, we need to handle error on request and on response by adding error listening during the request and response by logging the error into `stderr` using `request.on("error",..)` and `response.on("error,..)` including other things like testing codes, using try-catch, make sure the code works properly during development. Pro tip would be hiring a good coder so the error is prevented in the first place.