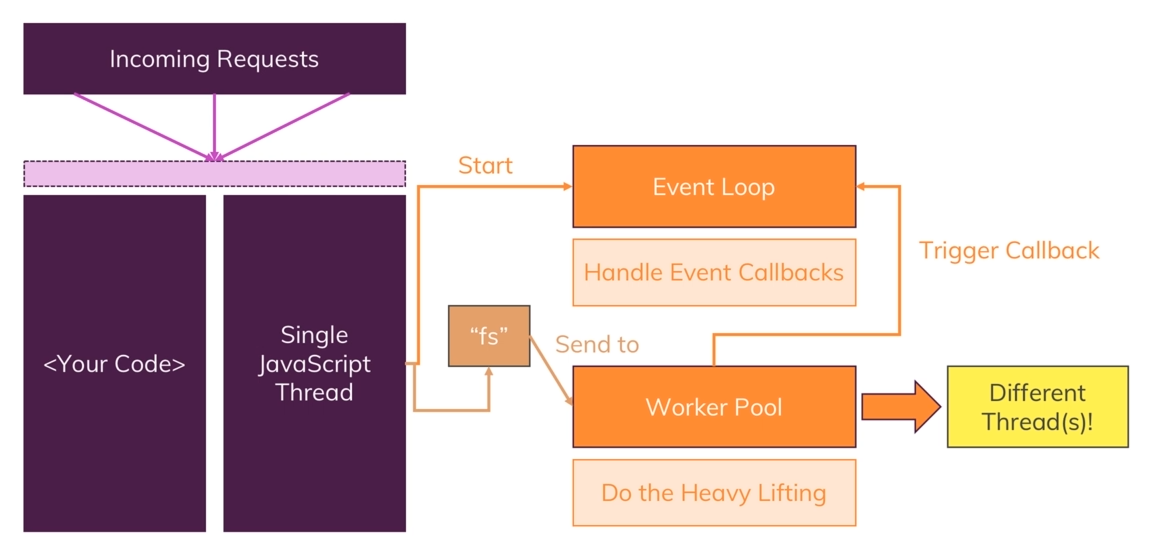
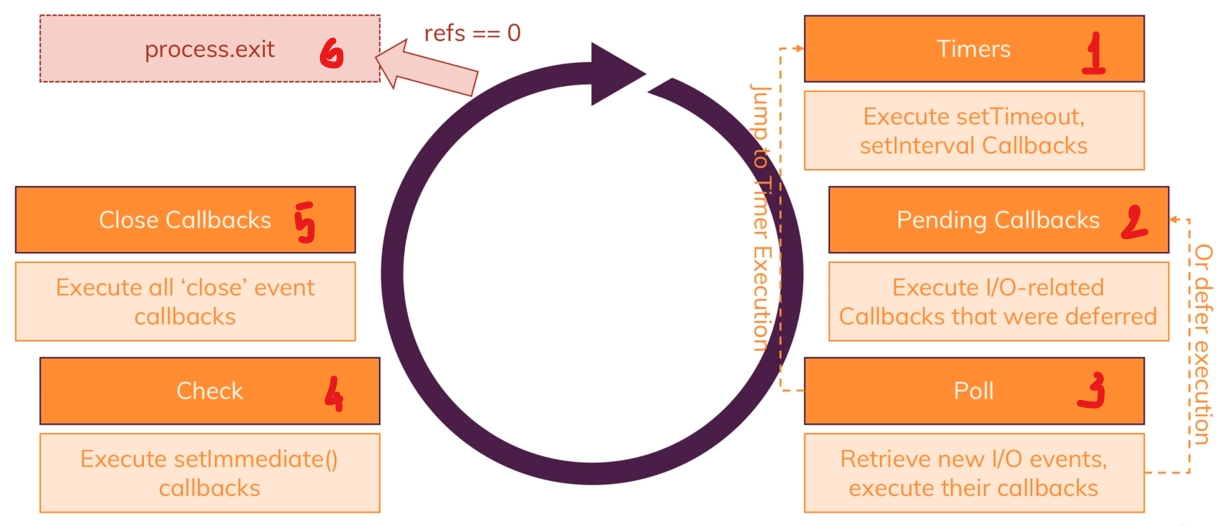
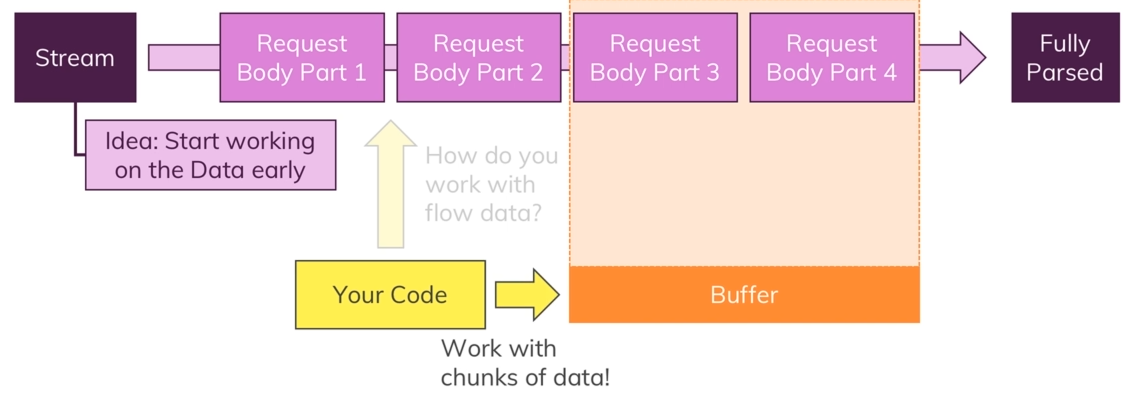
# Event Loop





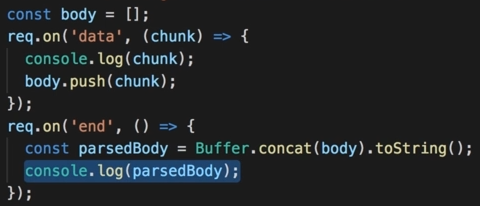
# Parsing Request Body



Nodejs nhận request body dưới dạng chunks vì nó không biết body lớn bao nhiêu

Vd ta send 1 file 1gb 🡪 rất to 🡪 việc chia thành nhiều chunks khiến ta dễ dàng xử lý nó

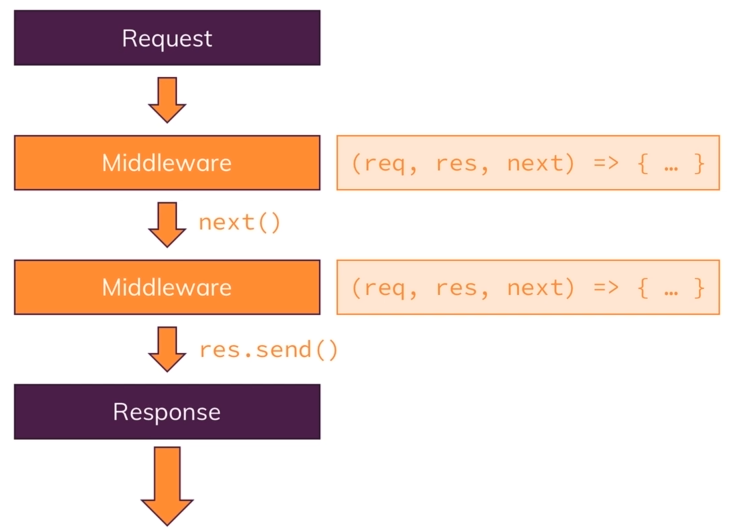
Buffer sẽ chứa ~ chunk này để thực hiện ops (lý do tại sao dùng buffer thì cần tìm hiểu thêm)

on(‘data’) sẽ listen khi request gửi data lên đến đâu thì nhận đến đó

on(‘end’) : kết thúc

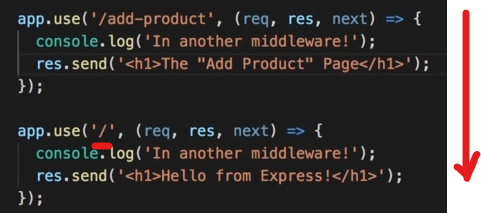
# Express

Được build dựa trên nguyên lý middleware



Default content-type : text/html

# Middleware Order



Thứ tự đặt middleware cũng rất quan trọng (từ trên xuống)

Với [ app.use(‘/’) ] thì nó sẽ match tất cả đường dẫn vì [/products, /index hay / ] thì đều bắt đầu với “/”

**Việc mình k gọi next() sẽ không chuyển req đến middleware tiếp theo**

\*nếu đã send res thì k gọi next() 🡪 tránh send res 2 lần

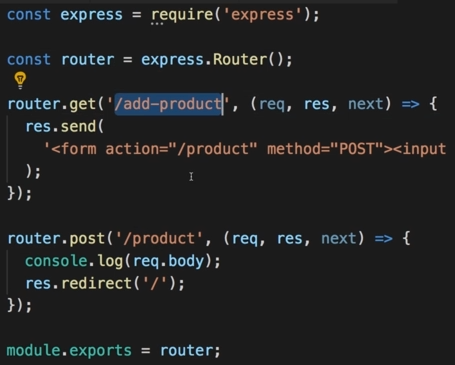


Body parser

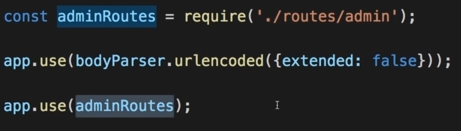


\*Nếu không use(bodyParser) 🡪 sẽ không nhận được gì ở req.body = undefined

# Router



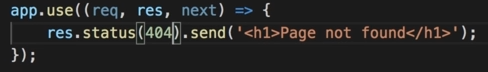
routes/admin.js



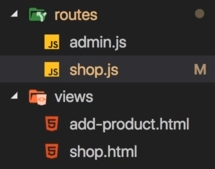
app.js

Lưu ý :

* Thứ tự quan trọng
* Matching của use() khác post() & get()
  + Post, get sẽ là exact match



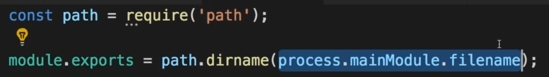
# Serve HTML



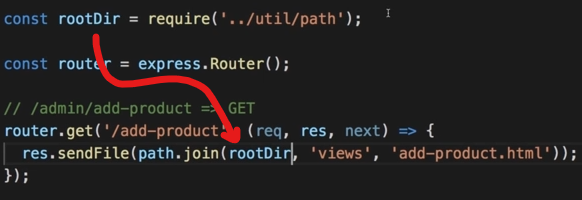


* \_\_dirname sẽ chỉ đến folder hiện tại của file which is /routes
  + Chính vì thế cần ../views/

# Cleaner Code với pathUtils



Utils/path.js



# Serve Static File

