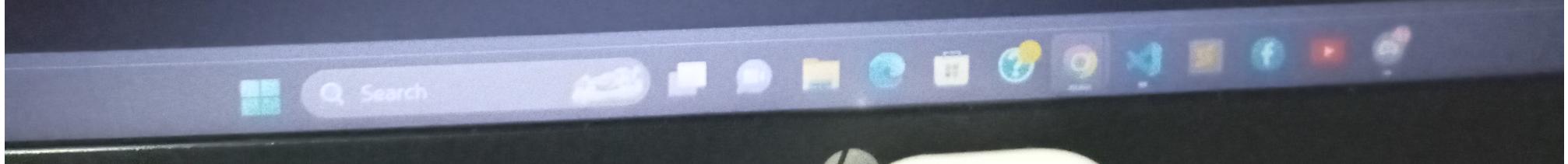


# Middleware via examples



# 1. If you go to an amusement park



Doesn't have ticket



# 1. If you go to an amusement park



Doesn't have ticket



# 1. If you go to an amusement park



Doesn't have ticket



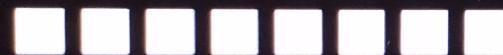
# 1. If you go to an amusement park

Gets kicked out



# 1. If you go to an amusement park

## Gets kicked out



# Middleware

# Official definition of middleware

## Using middleware

Express is a routing and middleware web framework that has minimal functionality of its own: An Express application is essentially a series of middleware function calls.

**Middleware** functions are functions that have access to the `request object (req)`, the `response object (res)`, and the next middleware function in the application's request-response cycle. The next middleware function is commonly denoted by a variable named `next`.

Middleware functions can perform the following tasks:

- Execute any code.
- Make changes to the request and the response objects.
- End the request-response cycle.
- Call the next middleware function in the stack.

<https://expressjs.com/en/guide/using-middleware.html>

# Official definition of middleware

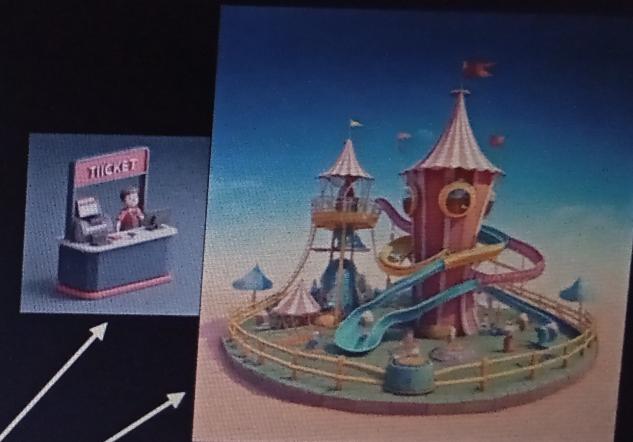
## Using middleware

Express is a routing and middleware web framework that has minimal functionality of its own: An Express application is essentially a series of middleware function calls.

**Middleware** functions are functions that have access to the `request object` (`req`), the `response object` (`res`), and the next middleware function in the application's request-response cycle. The next middleware function is commonly denoted by a variable named `next`.

Middleware functions can perform the following tasks:

- Execute any code.
- Make changes to the request and the response objects.
- End the request-response cycle.
- Call the next middleware function in the stack.



Middleware

b.100xdevs.com/courses/3/41

w3schools.com

devdocs.io

webarchive.org

M developer.mozilla.org

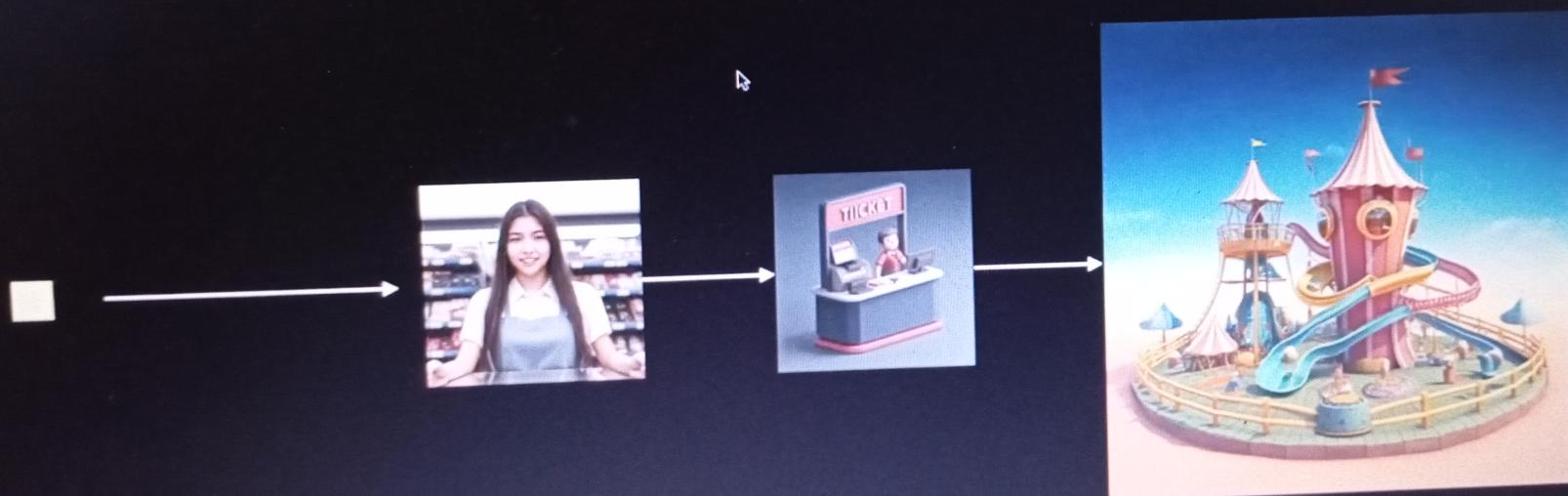
IO crop-circle.imageon...

Photobucket | My B...

Free Online Image...

Typing Practice

# An express app is essentially a series of middleware calls



X ▲ 100xdevs cohort X +

100xdevs.com/courses/3/41

v3schools.com devdocs.io webarchive.org developer.mozilla.org 10 crop-circle.imageon... Photobucket | My B... Free Online Image... Typing Practice

## Official definition of middleware

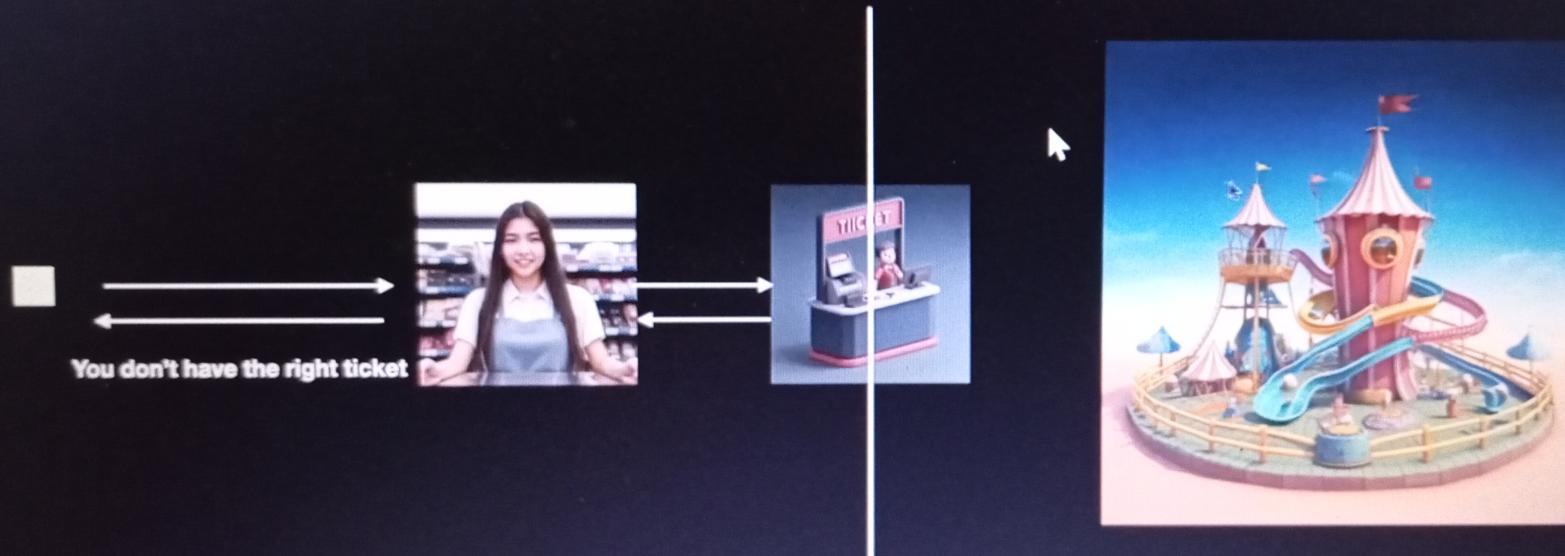
The diagram illustrates the concept of middleware. On the left, a woman is shown at a service counter with the text "You don't have the money". A vertical line with arrows at both ends connects this scene to a large, colorful roller coaster on the right. This visual metaphor represents how middleware acts as a layer between a user request and a final response, similar to how a ticket booth (middleware) processes requests before they reach the actual attraction (final destination).

Middleware functions can perform the following tasks:

- Execute any code
- Make changes to the request and the response objects
- End the request-response cycle
- Call the next middleware function in the stack

-42:03 1.5x ENG IN

## Official definition of middleware

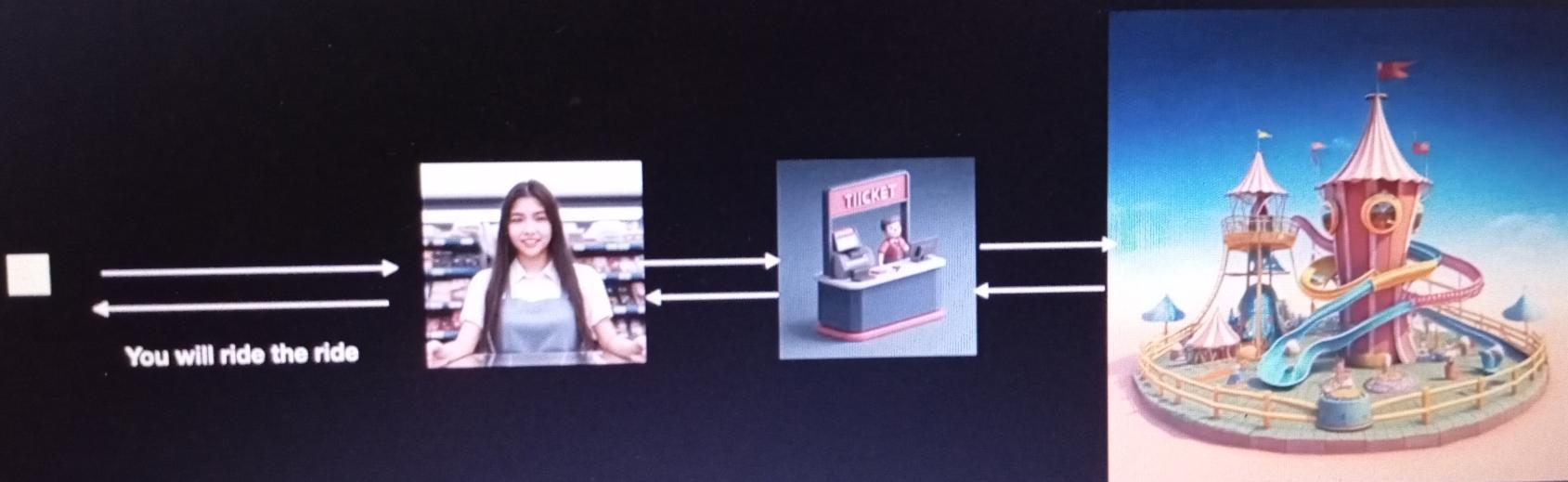


Middleware functions can perform the following tasks:

- Execute any code.
- Make changes to the request and the response objects.
- End the request-response cycle.
- Call the next middleware function in the stack.



# Official definition of middleware



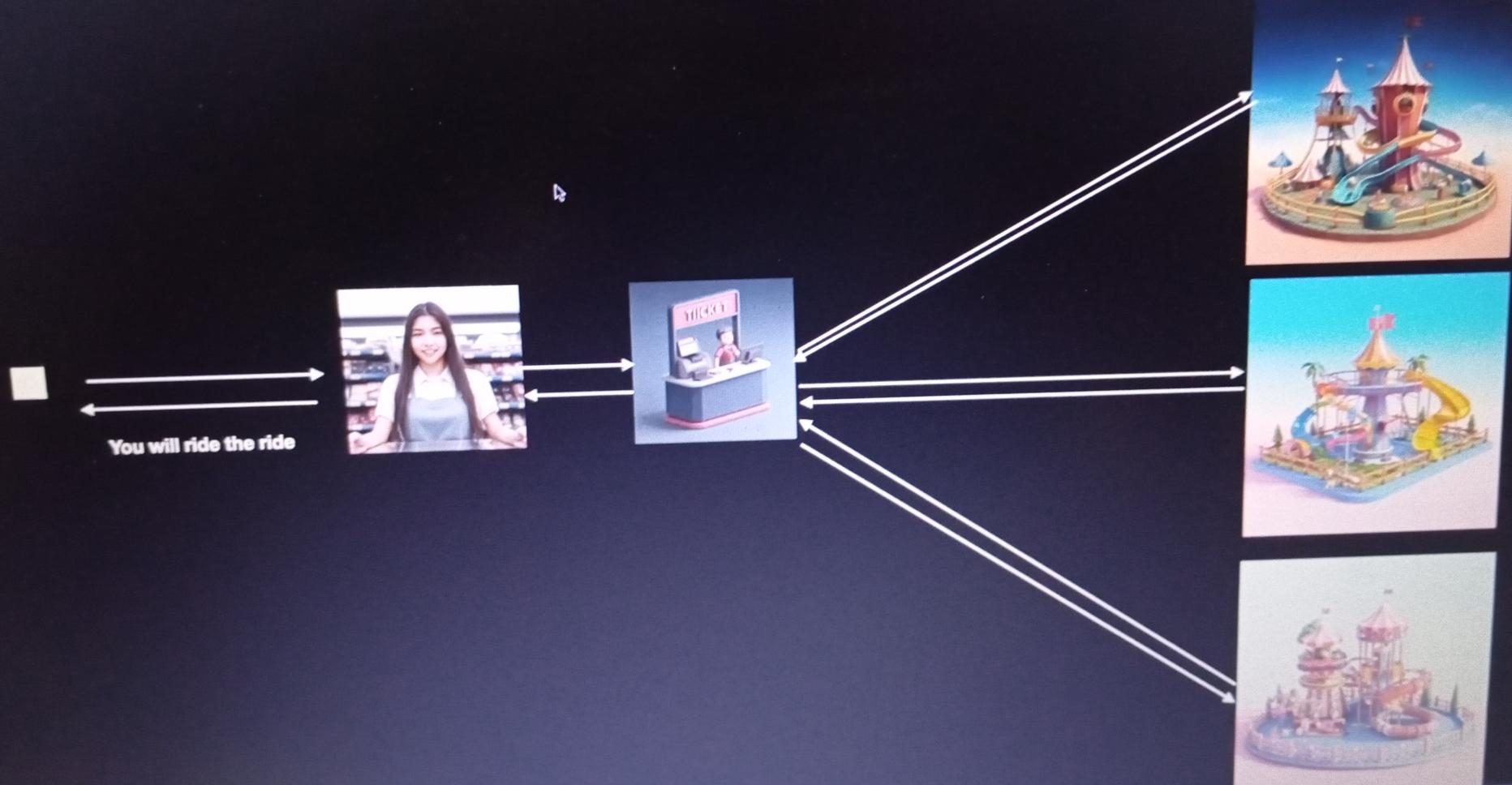
You will ride the ride

Middleware functions can perform the following:

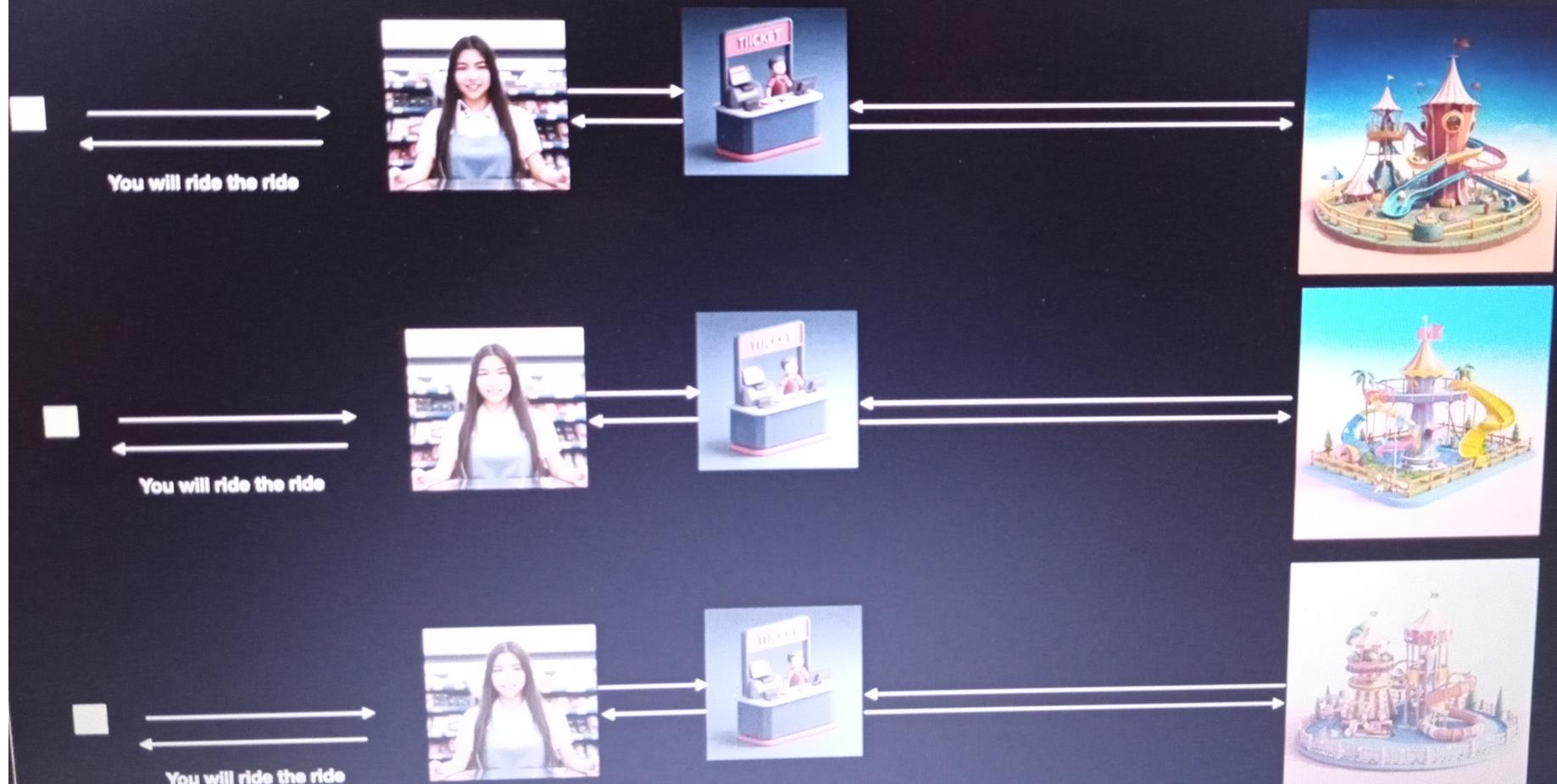
- Execute any code.
- Make changes to the request and response.
- End the request-response cycle.
- Call the next middleware function in the stack.

41.11

# Middlewares can be re-used

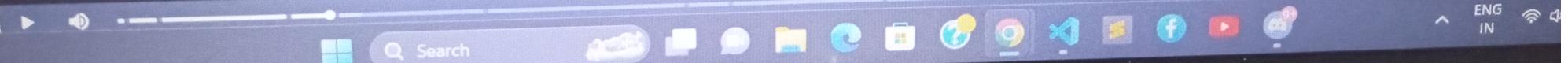


## This is bad (not re-using middleware)



-38.58 1.5x

ENG  
IN



# Lets codify this example

Ticket checker



Ride 1



Ride 2

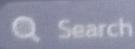


Ride 3



```
index.js ✘ × +  
index.js > ...  
1 const express = require("express");  
2 const app = express();  
3  
4 function ticketChecker(req, res, next) {  
5   const ticket = req.query.ticket;  
6   if (ticket === "free") {  
7     next();  
8   } else {  
9     res.status(403).send("Access denied");  
10  }  
11 }  
12  
13 app.use(ticketChecker);  
14  
15 app.get("/ride1", function () {  
16   res.send("You rode the first ride!");  
17 });  
18  
19 app.get("/ride2", function () {  
20   res.send("You rode the first ride!");  
21});  
22  
23 app.get("/ride3", function () {  
24   res.send("You rode the first ride!");  
25});  
26  
27 app.listen(3000);
```

<https://gist.github.com/hkirat/567e1ad5260a5fcac3fde209fabd34d4>



-3844

1.5x

ENG  
IN

07-0