

Middleware

It is an intermediary



in Express

Middleware in Express are functions that come into play after the server receives the request and before the response is sent to the client.

shenagupta7385@gmail.com

Middleware

Common middleware functions :

- methodOverride
- bodyParser
- express.static
- express.urlencoded


```
app.use(express.urlencoded({ extended: true }));
```

```
app.use(express.static(path.join(__dirname, "/public")));
```

- 1) req, res object
- 2) chaining
- 3) send a response



JS app.js

JS app.js >  app.get("/") callback

```
1  const express = require("express");
2  const app = express();
3
4  //middleware -> response send
5
6  app.get("/", (req, res) => {
7    |  res.send() | 
8  })
9
10 app.listen(8080, () => {
11   | console.log("server listening to port 8080");
12   | });
13
```

What do middlewares do?

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.**

Our 1st Middleware

app.use(middleware)

```
app.use(() => {  
  console.log("hi, I am a middleware");  
});
```

using req & res object in middleware

```
app.use((req, res) => {  
  console.log("hi, I am a middleware");  
  res.send("bye");  
});
```


JS app.js X

JS app.js > app.use() callback

```
1  const express = require("express");
2  const app = express();
3
4  app.use(() => {
5    | console.log("Hi, I am middleware");
6  });
7
8  app.get("/", (req, res) => {
9    | res.send("Hi, I am root.");
10  });
11
12  app.get("/random", (req, res) => {
13    | res.send("this is a random page");
14  });
15
16  app.listen(8080, () => {
17    | console.log("server listening to port 8080");
18  });
19
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
○ shradhakhapra@Shradhas-Air Middlewares % nodemon app.js
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
server listening to port 8080
Hi, I am middleware
```




Express 4.x - API Reference




New Incognito tab




← → × ⓘ localhost:8080

 Hoppscotch • Op...

 Mongoose

 Website

 AirBnb With Api I...

 & mongo

JS app.js ×

JS app.js > ...

```
1  const express = require("express");
2  const app = express();
3
4  app.use((req, res) => {
5    console.log("Hi, I am middleware");
6    res.send("middleware finished");
7  });
8  💡
9  app.get("/", (req, res) => {
10    res.send("Hi, I am root.");
11  });
12
13  app.get("/random", (req, res) => {
14    res.send("this is a random page");
15  });
16
17  app.listen(8080, () => {
18    console.log("server listening to port 8080");
19  });
```




localhost:8080

middleware finished



← → ↻ ⓘ localhost:8080/random

middleware finished

JS app.js >  app.use() callback

```
1  const express = require("express");
2  const app = express();
3
4  app.use((req, res) => {
5    let { query } = req.query;
6    console.log(query);
7    console.log("Hi, I am middleware");
8    res.send("middleware finished");
9  });
10
11 app.get("/", (req, res) => {
12   res.send("Hi, I am root.");
13 });
14
15 app.get("/random", (req, res) => {
16   res.send("this is a random page");
17 });
18
19 app.listen(8080, () => {
20   console.log("server listening to port 8080");
21 });
```

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

Hi, I am middleware

Hi, I am middleware

[nodemon] restarting due to changes...

[nodemon] starting `node app.js`

server listening to port 8080

[nodemon] restarting due to changes...

[nodemon] starting `node app.js`

server listening to port 8080

abcd

Hi, I am middleware





Using next


The next middleware function is commonly denoted by a variable named **next**.

```
app.use((req, res, next) => {  
  console.log('Time:', Date.now())  
  next()  
})
```

If the current middleware function does not end the request-response cycle, it must call `next()` to pass control to the next middleware function.

JS app.js X

JS app.js >  app.use() callback

```
1  const express = require("express");
2  const app = express();
3
4  app.use((req, res, next) => {
5     console.log("Hi, I am middleware");
6    next();
7  });
8
9  app.get("/", (req, res) => {
10   res.send("Hi, I am root.");
11 });
12
13 app.get("/random", (req, res) => {
14   res.send("this is a random page");
15 });
16
17 app.listen(8080, () => {
18   console.log("server listening to port 8080");
19 });
```




Using Express middleware



localhost:8080/random



localhost:8080/random

this is a random page



ex Using Express middleware




localhost:8080



localhost:8080

Hi, I am root.



JS app.js >  app.use() callback

```
1  const express = require("express");
2  const app = express();
3
4  app.use((req, res, next) => {
5    console.log("Hi, I am 1st middleware");
6    next();
7  });
8
9  app.use((req, res, next) => {
10   console.log("Hi, I am 2nd middleware");
11   next();
12 });
13
14 app.get("/", (req, res) => {
15   res.send("Hi, I am root.");
16 });
17
18 app.get("/random", (req, res) => {
19   res.send("this is a random page");
20 });
```

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

server listening to port 8080

Hi, I am middleware

Hi, I am middleware

[nodemon] restarting due to changes...


[nodemon] starting `node app.js`

server listening to port 8080

Hi, I am 1st middleware

Hi, I am 2nd middleware

JS app.js X

JS app.js >  app.use() callback

```
1  const express = require("express");
2  const app = express();
3
4  app.use((req, res, next) => {
5    console.log("Hi, I am 1st middleware");
6    next();
7    console.log("this is after next()");
8  });
9
10 app.use((req, res, next) => {
11   console.log("Hi, I am 2nd middleware");
12   next();
13 });
14
15 app.get("/", (req, res) => {
16   res.send("Hi, I am root.");
17 });
```

PROBLEMS


OUTPUT


TERMINAL

DEBUG CONSOLE

```
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
server listening to port 8080
Hi, I am 1st middleware
Hi, I am 2nd middleware
[nodemon] restarting due to changes...
[nodemon] starting `node app.js`
server listening to port 8080
Hi, I am 1st middleware
Hi, I am 2nd middleware
this is after next()
```

□

JS app.js >  app.use() callback

```
1  const express = require("express");
2  const app = express();
3
4  app.use((req, res, next) => {
5    console.log("Hi, I am 1st middleware");
6    return next();
7    console.log("abc");
8  });
9  
10 app.use((req, res, next) => {
11   console.log("Hi, I am 2nd middleware");
12   next();
13 });
14
15 app.get("/", (req, res) => {
16   res.send("Hi, I am root.");
```

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

```
[nodemon] restarting due to changes...
```

```
[nodemon] starting `node app.js`
```

```
server listening to port 8080
```

```
Hi, I am 1st middleware
```

```
Hi, I am 2nd middleware
```

```
this is after next()
```

```
[nodemon] restarting due to changes...
```

```
[nodemon] starting `node app.js`
```

```
server listening to port 8080
```

```
[nodemon] restarting due to changes...
```

```
[nodemon] starting `node app.js`
```

```
server listening to port 8080
```

```
Hi, I am 1st middleware
```

```
Hi, I am 2nd middleware
```

```
□
```

Creating **Utility** Middleware

Logger

```
app.use((req, res, next) => {  
  req.responseTime = new Date(Date.now()).toString();  
  console.log(req.method, req.path, req.responseTime, req.hostname);  
  next();  
});
```

Creating **Utility** Middleware

Logger → log (useful info console print)

```
app.use((req, res, next) => {  
  req.responseTime = new Date(Date.now()).toString();  
  console.log(req.method, req.path, req.responseTime, req.hostname);  
  next();  
});
```


JS app.js

snehagupta/385@gmail.com

JS app.js > app.use() callback

```
7 // });  
8  
9 // app.use((req, res, next) => {  
10 //   console.log("Hi, I am 2nd middleware");  
11 //   next();  
12 // });  
13  
14 app.use((req, res, next) => {  
15   console.log(req.method, req.hostname, req.path);  
16   next();  
17 });  
18  
19 app.get("/", (req, res) => {
```

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

GET

GET

POST

PATCH

[nodemon] restarting due to changes...

[nodemon] starting `node app.js`

server listening to port 8080

PATCH localhost

GET localhost

[nodemon] restarting due to changes...

[nodemon] starting `node app.js`

server listening to port 8080

GET localhost undefined

[nodemon] restarting due to changes...

[nodemon] starting `node app.js`

server listening to port 8080

GET localhost /

GET localhost /random

... JS app.js X


JS app.js > ...

```
11 // next();
12 // });
13
14 app.get("/", (req, res) => {
15   res.send("Hi, I am root.");
16 });
17
18 app.get("/random", (req, res) => {
19   res.send("this is a random page");
20 });
21 💡
22 app.use((req, res, next) => {
23   req.time = Date.now();
24   console.log(req.method, req.hostname, req.path, req.time);
25   next();
26 });
27
28 app.listen(8080, () => {
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
○ shradhakhapra@Shradhas-Air Middlewares % nodemon app.js
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
server listening to port 8080
□
```

JS app.js ×

JS app.js >  app.use("/random") callback

```
8
9 // app.use((req, res, next) => {
10 //   console.log("Hi, I am 2nd middleware");
11 //   next();
12 // });
13
14 app.use("/random", (req, res, next) => {
15 |   console.log("I am only for random");
16 |   next();
17 | });
18
19 app.get("/", (req, res) => {
20 |   res.send("Hi, I am root.");
21 | });
22
23 app.get("/random", (req, res) => {
24 |   res.send("this is a random page");
25 | });
```

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

```
○ shradhakhapra@Shradhas-Air Middlewares % nodemon app.js
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
server listening to port 8080
```

Express 4.x - API Reference

localhost:8080

localhost:8080

Hi, I am root.

ex Express 4.x - API Reference



localhost:8080



localhost:8080/random|



localhost:8080/random



localhost:8080/random - Google Search

Hi, I am root.

JS app.js



JS app.js > app.use("/random") callback

```
8
9 // app.use((req, res, next) => {
10 //   console.log("Hi, I am 2nd middleware");
11 //   next();
12 // });
13
14 app.use("/random", (req, res, next) => {
15   console.log("I am only for random");
16   next();
17 });
18
19 app.get("/", (req, res) => {
20   res.send("Hi, I am root.");
21 });
22
23 app.get("/random", (req, res) => {
24   res.send("this is a random page");
25 });
```

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

```
○ shradhakhapra@Shradhas-Air Middlewares % nodemon app.js
[nodemon] 3.0.1
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
server listening to port 8080
I am only for random
```



Express 4.x - API Reference



Error



localhost:8080/random/something

Cannot GET /random/something

JS app.js ×

JS app.js > app.use("/random") callback

8

9 // app.use((req, res, next) => {

10 // console.log("Hi, I am 2nd middleware");

11 // next();

12 // });

13

14 app.use("/rand

15 console.log(

16 next();

17 });

18

19 app.get("/", (req, res) => {

20 res.send("Hi, I am root.");

21 });

22

23 app.get("/random", (req, res) => {

24 res.send("this is a random page");

25

'req' is declared but its value is never re

(parameter) req: Request<{}, any, any, qs.R

any>>

Quick Fix... (⌘.)

PROBLEMS

OUTPUT

TERMINAL

DEBUG CONSOLE

○ shradhakhapra@Shradhas-Air Middlewares % nodemon app.js

[nodemon] 3.0.1

[nodemon] to restart at any time, enter `rs`

[nodemon] watching path(s): *.*

[nodemon] watching extensions: js,mjs,cjs,json

[nodemon] starting `node app.js`

server listening to port 8080

I am only for random

I am only for random

GET localhost /random/something 1695888487737

app.use Callback

apis
/api / listings ✓
/api / listings /: id ✓

REST
/ listings ✓
/ listings /: id ✓

app.use Callback

apis

/api / listings ✓

/api / listings /: id ✓

/api

middleware



login

REST

/ listings ✓

/ listings /: id ✓

JS app.js



snehagupta7385@gmail.com

JS app.js > ...

```
23 app.get("/random", (req, res) => {
24   res.send("this is a random page");
25 });
26
27 //logger - morgan
28 app.use((req, res, next) => {
29   req.time = Date.now();
30   console.log(req.method, req.hostname, req.path, req.time);
31   next();
32 });
33
34 //404
35 app.use((req, res) => {
36   res.send("Page not found!");
37 });
38
39 app.listen(8080, () => {
40   console.log("server listening to port 8080");
```

Express 4.x - API Reference

localhost:8080/abcd

localhost:8080/abcd

Page not found!

API Token as Query String

Let's create a middle-ware for an api that checks if the access token was passed in the query string or not.

API Token as Query String


Let's create a middle-ware for an api that checks if the access token was passed in the query string or not.


/api
↓
res ⇒ data

req
/api ? token = give access

for apis
↓
access token

JS app.js X

JS app.js >  app.use("/api") callback

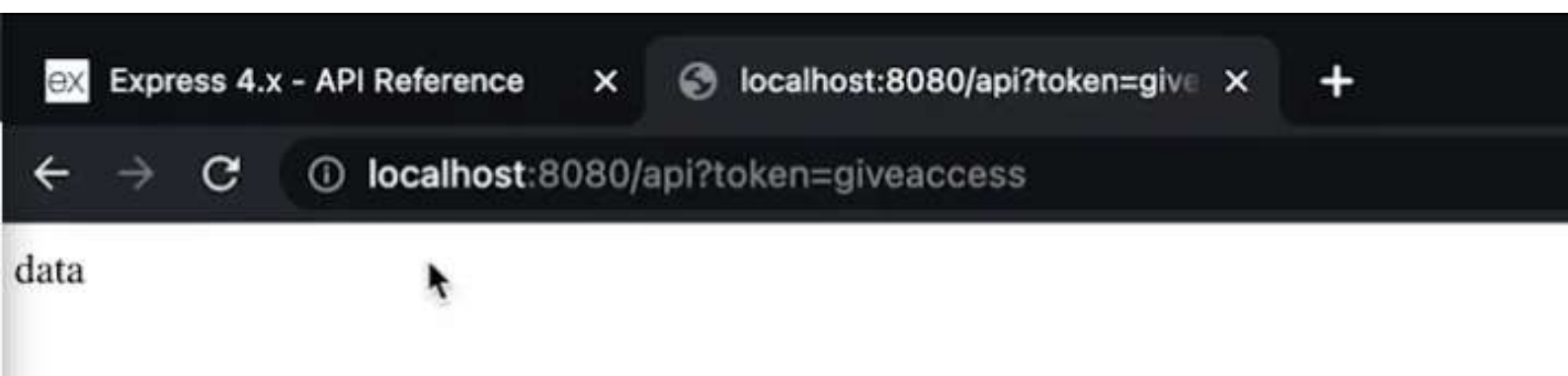
```
10 // console.log("Hi, I am 2nd middleware");
11 // next();
12 // });
13 
14 app.use("/api", (req, res, next) => {
15   let { token } = req.query;
16   if (token === "giveaccess") {
17     next();
18   }
19   res.send("ACCESS DENIED!");
20 });
21
22 app.get("/api", (req, res) => {
23   res.send("data");
24 });
25
26 app.get("/", (req, res) => {
```

Express 4.x - API Reference

localhost:8080/api

localhost:8080/api

ACCESS DENIED!





Express 4.x - API Reference



localhost:8080/api?token=give



localhost:8080/api?token=giveaccesscxvaszxc

ACCESS DENIED!

JS app.js

JS app.js > ...

```
11 // next();
12 // });
13
14 const checkToken = (req, res, next) => {
15   let { token } = req.query;
16   if (token === "giveaccess") {
17     next();
18   }
19   res.send("ACCESS DENIED!");
20 };
21
22 app.get("/api", checkToken, (req, res) => {
23   res.send("data");
24 });
25
26 app.get("/", (req, res) => {
27   res.send("Hi, I am root.");
28 });
```

Handling Errors

Express **Default** Error Handler

Syntax

class

snehagupta7385@gmail.com
dev → production

JS app.js ×

JS app.js > [🔍] checkToken

```
1 //    // });  
2  
3  
4  
5  
6  
7  
8  
9 // app.use((req, res, next) => {  
10 //   console.log("Hi, I am 2nd middleware");  
11 //   next();  
12 // });  
13  
14 const checkToken = (req, res, next) => {  
15   let { token } = req.query;  
16   if (token === "giveaccess") {  
17     next();  
18   }  
19   throw new Error("ACCESS DENIED!");  
20 };  
21
```

```
Error: ACCESS DENIED!
    at checkToken (/Users/shradhakhapra/WebClassroom/Backend/Middlewares/app.js:19:9)
    at Layer.handle [as handle_request] (/Users/shradhakhapra/WebClassroom/Backend/Middlewares/node_modules/express/lib/router/layer.js:95:5)
    at next (/Users/shradhakhapra/WebClassroom/Backend/Middlewares/node_modules/express/lib/router/route.js:144:13)
    at Route.dispatch (/Users/shradhakhapra/WebClassroom/Backend/Middlewares/node_modules/express/lib/router/route.js:114:3)
    at Layer.handle [as handle_request] (/Users/shradhakhapra/WebClassroom/Backend/Middlewares/node_modules/express/lib/router/layer.js:95:5)
    at /Users/shradhakhapra/WebClassroom/Backend/Middlewares/node_modules/express/lib/router/index.js:284:15
    at Function.process_params (/Users/shradhakhapra/WebClassroom/Backend/Middlewares/node_modules/express/lib/router/index.js:346:12)
    at next (/Users/shradhakhapra/WebClassroom/Backend/Middlewares/node_modules/express/lib/router/index.js:280:10)
    at expressInit (/Users/shradhakhapra/WebClassroom/Backend/Middlewares/node_modules/express/lib/middleware/init.js:40:5)
    at Layer.handle [as handle_request] (/Users/shradhakhapra/WebClassroom/Backend/Middlewares/node_modules/express/lib/router/layer.js:95:5)
```