

# Middleware



## Why middleware?

If you want "something" to happen before your router calls are called, you use middleware.

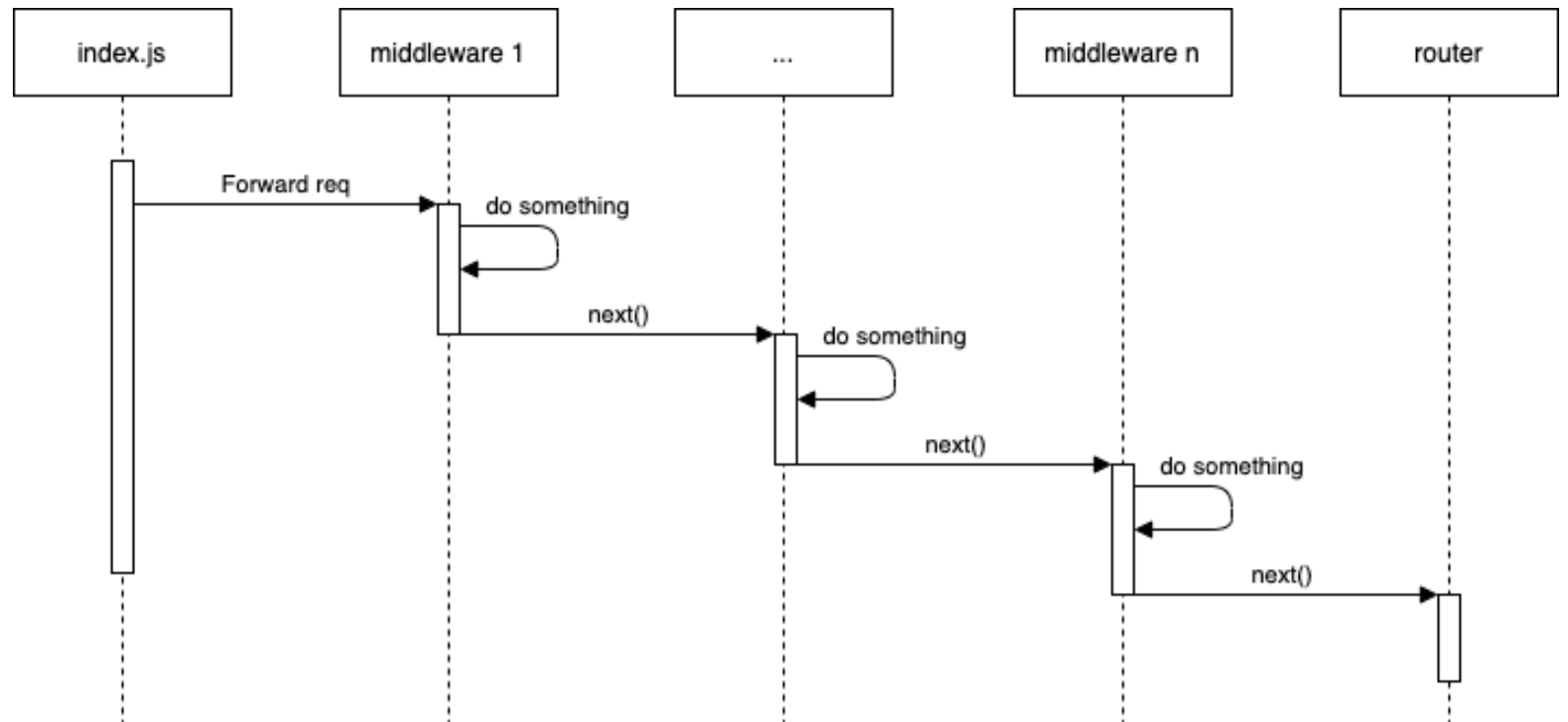
Two scenarios:

1. Before **every** call – global middleware
2. Before **some** calls – router middleware

# Middleware: Workings

Middleware works with the `next()` function.

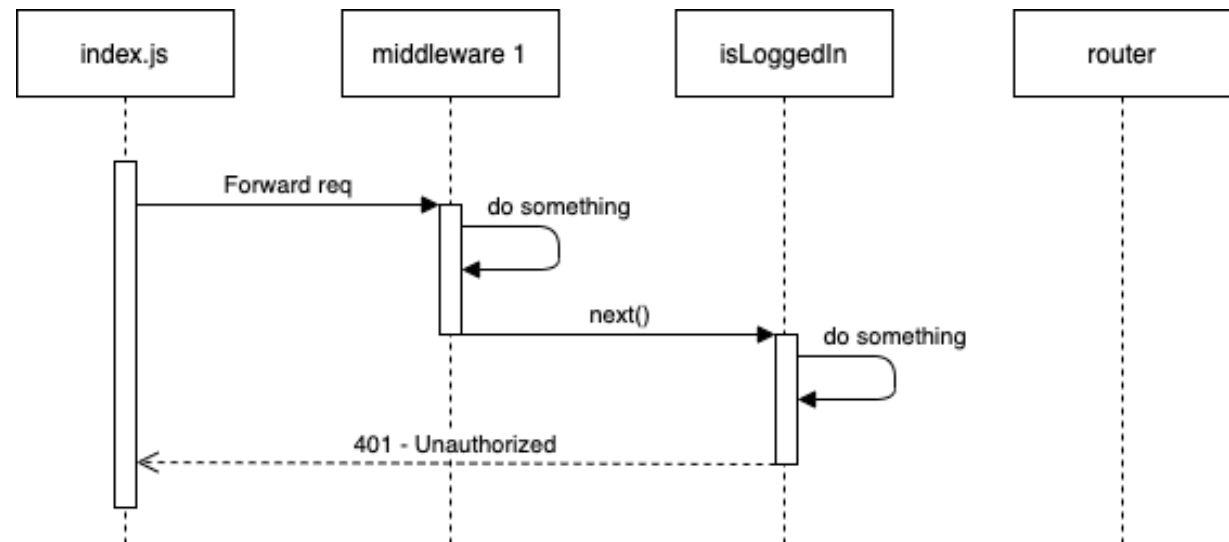
```
const myFirstMiddleware = (req, res, next) => {  
  // Do your stuff here  
  next();  
}
```



# Middleware: Workings

## Middleware can return a response early.

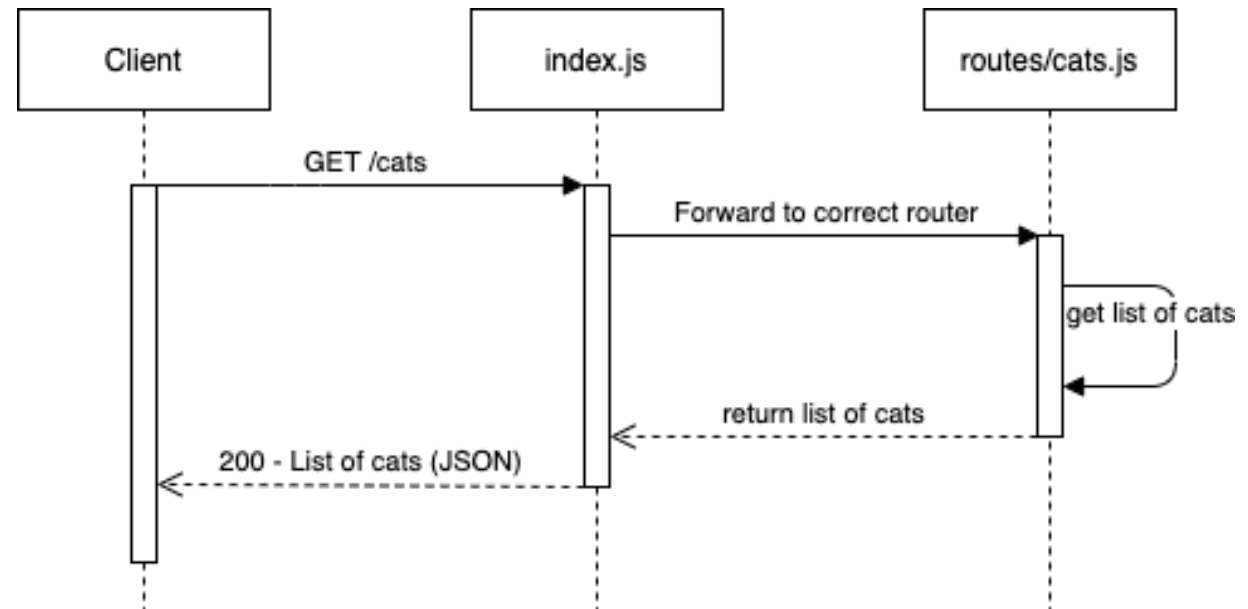
```
const isLoggedIn = (req, res, next) => {  
  // Check if user is logged in  
  
  // Yes?  
  next();  
  
  // No?  
  res.status(401).json({msg: "You are not logged in!"});  
}
```



# Middleware: Scenario

Our routers are configured as follows:

```
import catsRouter from "../routes/cats.js";  
app.use('/cats', catsRouter);  
  
// routes/cats.js  
router.get('', (req, res) => {  
  res.json(cats);  
});
```

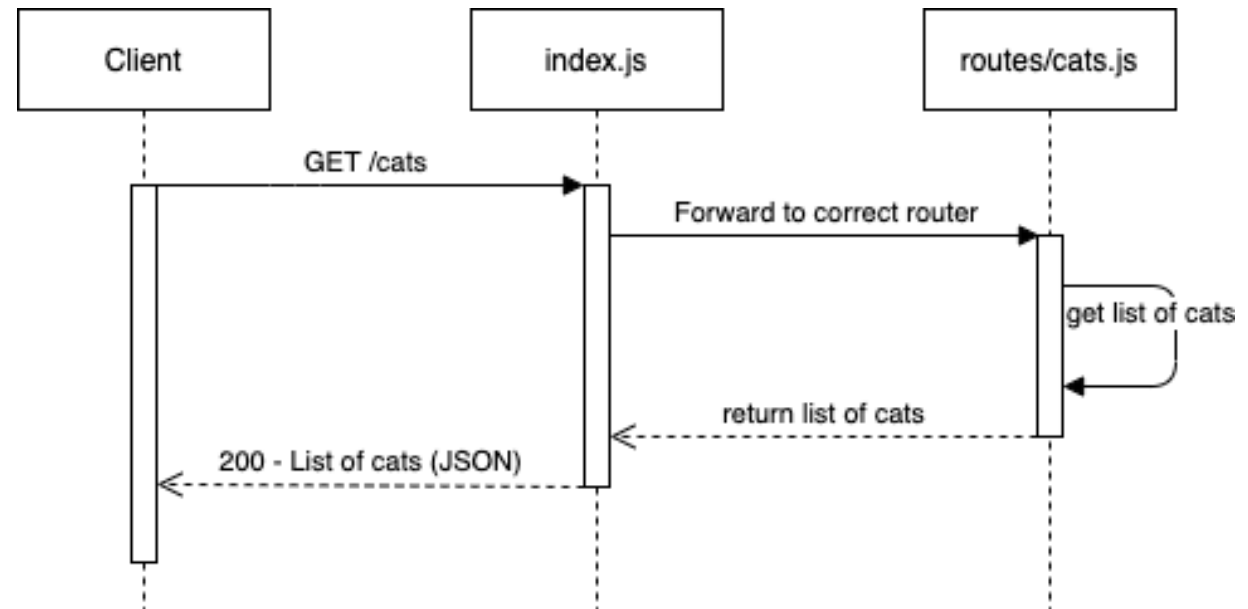


# Middleware: Goal

We want to print the current time everytime a request is made.

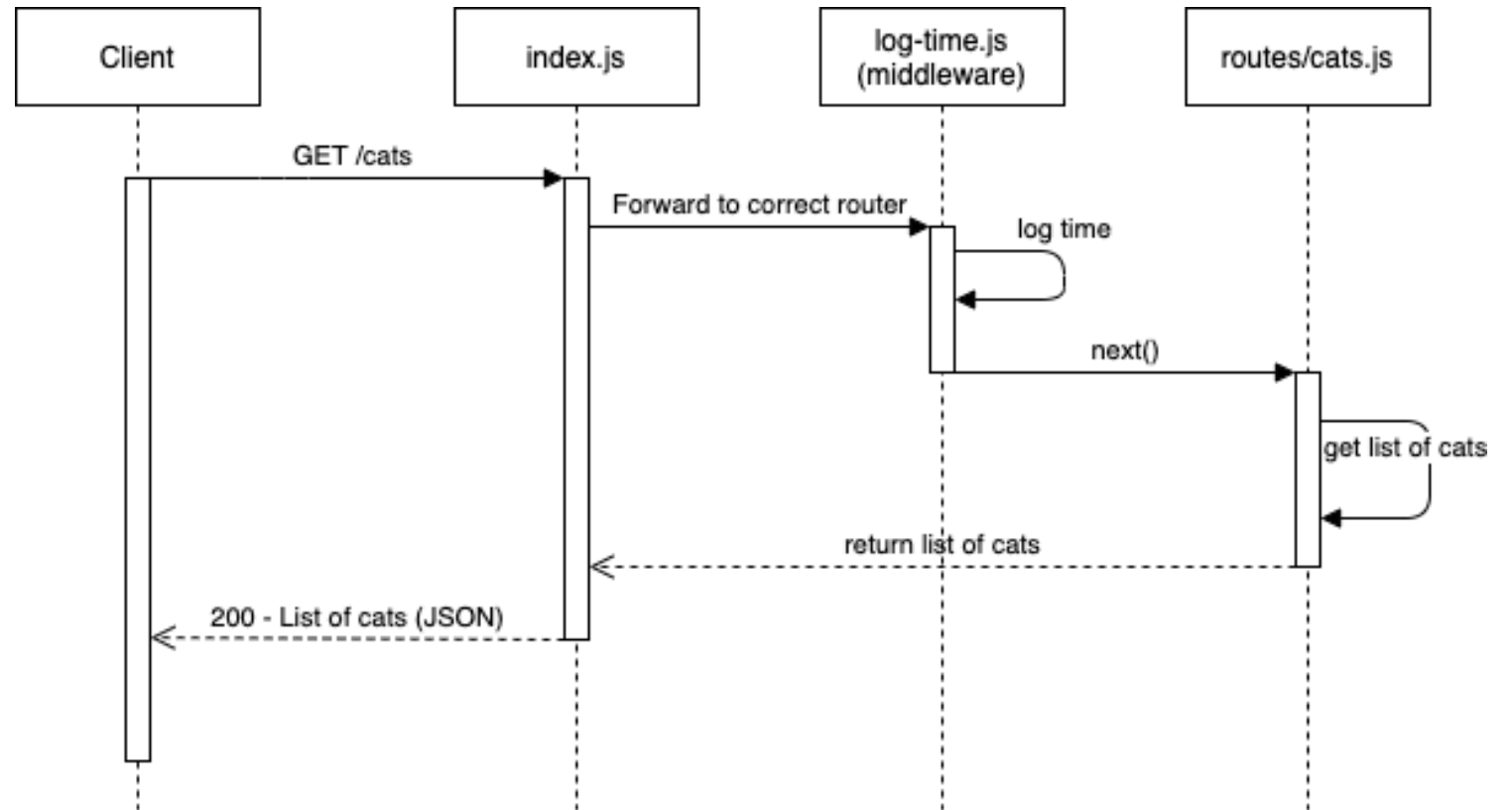
```
import catsRouter from "../routes/cats.js";  
app.use('/cats', catsRouter);
```

```
// routes/cats.js  
router.get('/', (req, res) => {  
  res.json(cats);  
});
```



# Middleware: Solution

We want to print the current time  
everytime a request is made.



# Middleware: Solution

We want to print the current time  
everytime a request is made.

```
// Middleware/log-time.js  
const logTime = (req, res, next) => {  
    console.log('Request time', new Date());  
    next();  
};
```

```
// index.js ... Not shown for brevity
```

```
// Middleware  
app.use(express.json());  
app.use(logTime);
```

```
// Routes  
app.use('/cats', catsRouter);
```



# Middleware: Only some paths

You can also register middleware for only some paths

```
import isLoggedIn from "../middleware/is-logged-in.js";

router.get('', (req, res) => {
  res.json(cats);
});

router.post('', isLoggedIn, (req, res) => {
  // Do your thing
});
```

# Questions?

# Assignment:

Work on your project