20/03/2024

Express Middlewares

```
//create a middleware --> (res,res,next)=>{}
const middleware1 = (req,res,next)=>{
    console.log("middleware-1 executed");
    //res.send({message:"Reply from middleware-1"});
    next(); //goes to the next middleware or response
}

const middleware2 = (req,res,next)=>{
    console.log("middlware-2 executed");
    //res.send({message:"Reply from middleware-2"});
    next();


}

const middleware3 = (req,res,next)=>{
    console.log("middleware-3 executed");
    //res.send({message:"Reply from middleware-3"});
    next();
}
//use as app-level middleware --> use()
app.use(middleware1); //calling the middlewares
app.use(middleware3);
app.use(middleware2);
```

Explanation :

The order of use is Middleware 1,3,2 as observed from the code

* So, first the middleware-1 is executed, then checks if any response code is there. Here, I commented the line res.send in middleware. If it is not commented, then the next() line of the code is executed.

* Here, the next middleware is executed i.e., middleware-3 as in the order of execution. Again, it checks if there is any response code. If there, then the response is given and the execution stops. Else, the next() is executed.

* Now, in the order is the next middleware i.e., middleware-2. Same again. It checks for the response code and executes it if exists or the next() line.

* Now, here we do not have any more middlewares in the order of execution. So, now the request is handled. As observed, middlewares get executed when a certain request is made. Say, we made a GET request. First, the middlewares 1,3,2 are executed as discussed above and then the GET request is handled and we get our requested Data as output.

Application Level Middlewares vs Route Level Middlewares

```
app.use(middleware1); //App Level Middlewares
app.use(middleware3);

app.post('/user',middleware2,(req,res)=>{
//Route-level middleware
    //get new user from request
    let newUser = req.body;
    console.log(newUser);
    //push new user to users array
    users.push(newUser);
    res.send({message:"New User Created"});
})
```

From this code, we can observe that, app-level middlewares - 1,3 are executed for every request but, the middleware-2 is executed only for the POST request.

So, Middlewares 1,3 are app-level and middleware - 2 is route level.

* Middleware Syntax : (req,res,next) => {}
* Error Handling Middleware Syntax : (err,req,res,next) => {}

* Error Handling Middleware should always be at the last of all the code, else it executes for every line of code. It automatically catches the error if occurred at any of the routes and also we don't need to call it.