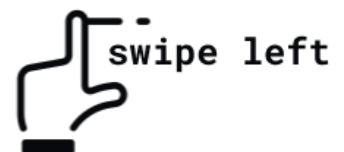
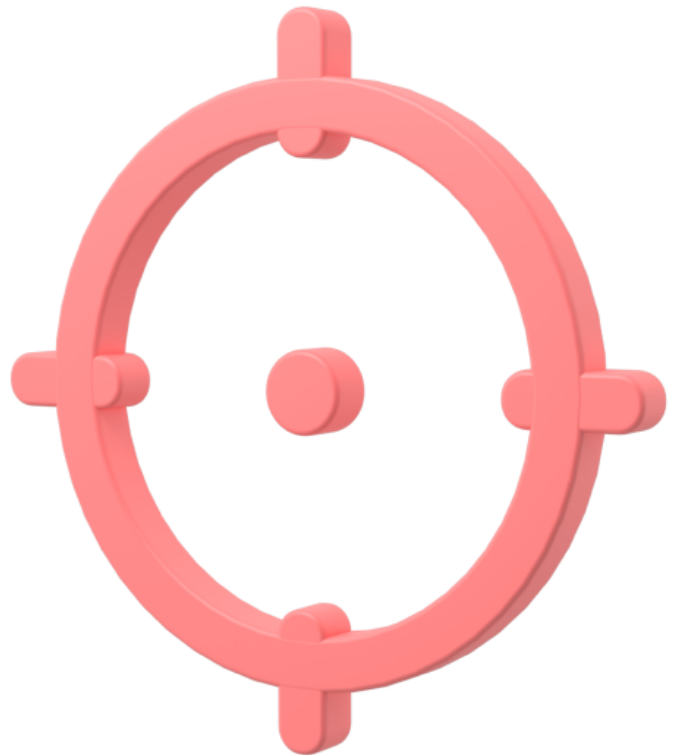


MIDDLEWARE IN EXPRESS



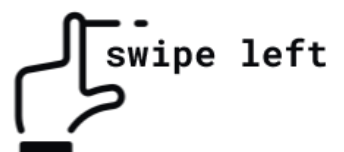
MIDDLEWARE IN EXPRESS

Express is a Node.js framework that provides many features for building server side applications.

Middleware is a concept used in many languages and tools, and also in express. What does it mean and how to implement it?

As the name already gives, it is something that **happens in the middle**. Think of middlewares as a intermediary. They are middle functions that determine if the next function would be called or not.

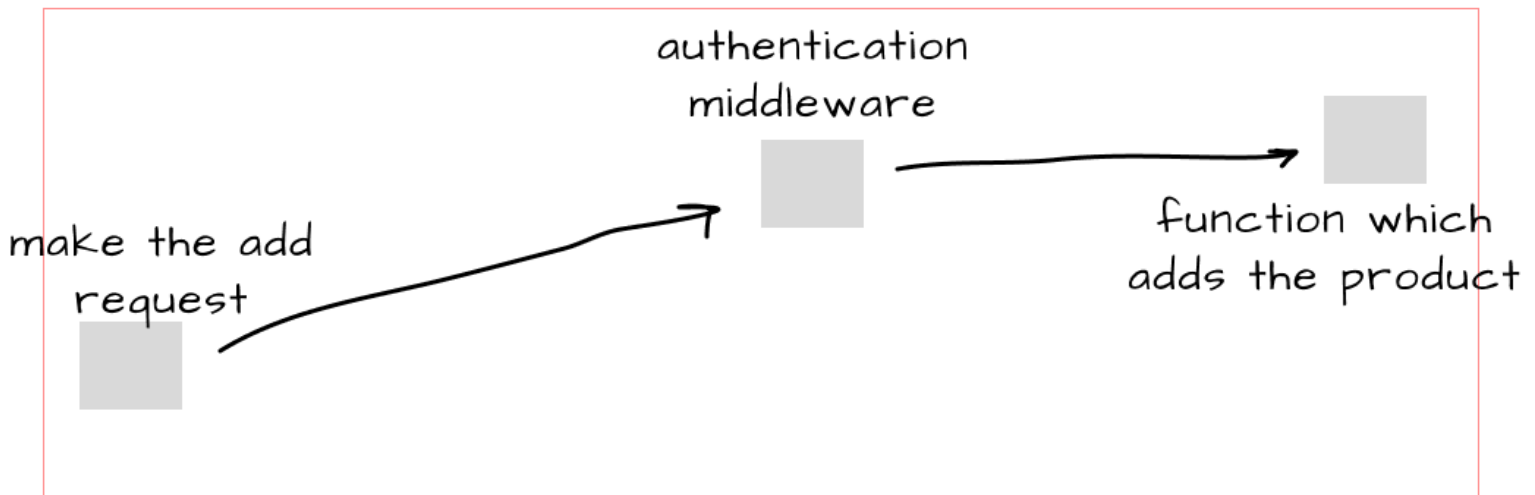
When you think of this in applications, middlewares can be applied to validate data, authenticate user, and so on, before the next function is called.



MIDDLEWARE IN EXPRESS

For example, you create an API for adding products to your backend database. You can create an **authentication middleware** which checks that the user trying to add that product has valid credentials.

If the user indeed has valid credentials, the middleware passes the request to the next function which adds the product. And if the user does not, the middleware returns an error and the next function will not be called.



MIDDLEWARE IN EXPRESS

Now, I'll show you a simple Express Application and how to create a middleware.

```
const express = require("express")

const app = express()

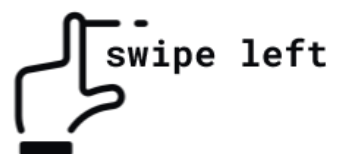
function getMyProducts(req, res) {
  res.send("my products")
}

app.get("/my_products", getMyProducts)

app.listen(8000, function() {
  console.log("Server started!")
})
```

This is a simple server with one API route: **/my_products**.

When you go to this route, the **getMyProducts** function is called with the request and response argument. Let's add a middleware...



MIDDLEWARE IN EXPRESS

```
const express = require("express")

const app = express()

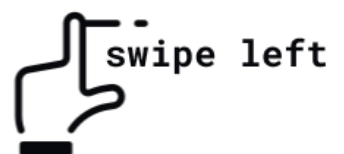
function isLoggedIn(req, res, next) {
  // check if user is logged in
  if (req.isLoggedIn) {
    next()
  } else {
    res.status(400).send("User is not logged in")
  }
}

function getMyProducts(req, res) {
  res.send("my products")
}

app.get("/my_products", isLoggedIn, getMyProducts)

app.listen(8000, function() {
  console.log("Server started!")
})
```

Explanation of code in next image...



MIDDLEWARE IN EXPRESS

In the code on the left image, I created an **isLoggedIn** middleware. This middleware receives the **request**, **response** and **next** argument.

In this middleware, I check if the user is logged in. If the user is, I call **next()**, which means **proceed to the next function that handles the API**. If the user isn't, I return an error.

And for the API declaration, you can see I did
(“/_products”, isLoggedIn, getMyProducts)

This way, I've specified **isLoggedIn** to be a middleware, which will pass to **getMyProducts**, the next function, if the user is logged in.