[Khaled Abdelrahman] [Link]
[Ibrahim Hafez Abdallah][Link]
[Gerges Zechariah][Link]
[Team]

What is middleware?

By the definition of Microsoft Middleware is software that lies between an operating system and the applications running on it. Essentially functioning as a hidden translation layer, middleware enables communication and data management for distributed applications. It's sometimes called plumbing, as it connects two applications together so data and databases can be easily passed between the "pipe." Using middleware allows users to perform such requests as submitting forms on a web browser, or allowing the web server to return dynamic web pages based on a user's profile.

So, in simple words it's a type of computer software that provides services to software applications beyond those available from the operating system. It can be described as "software glue.

Middleware makes it easier for software developers to implement communication and input/output, so they can focus on the specific purpose of their application. It gained popularity in the 1980s as a solution to the problem of how to link newer applications to older legacy systems, although the term had been in use since 1968.

What does middleware do in node.js?

The middleware in node.js is a function that will have all the access for requesting an object, responding to an object, and moving to the next middleware function in the application request-response cycle. This function can be used for modifying the req and res objects for tasks like adding response headers, parsing requesting bodies, and so on.

SO, as name suggests it comes in middle of something and that is request and response cycle

-Middleware has access to request and response object

-Middleware has access to next function of request-response life cycle

Why is middleware needed?

Middleware is required for helping developers in building applications more effectively and efficiently. The middleware will act as a connection between data, applications, and the users. When you are an organization with a multi-cloud environment, then middleware will make it more cost-effective for the development and running of the application at scale.

Middleware products and providers

Middleware programs come in on-premises software and cloud services; they can be used independently or together, depending on the use case. While cloud providers bundle middleware into cloud services suites -- such as middleware as a service (MWaaS) or integration platform as a service (iPaaS) -- many businesses may choose independent middleware products that fit their specific needs.

Some vendors that offer middleware include:

· IBM with IBM Worklight, which is used for developing cross-platform mobile applications.

· Microsoft with Microsoft BizTalk, which is used for integration hubs.

· SAP with SAP NetWeaver Mobile for mobile applications.

· Apache with Apache Camel, which provides an open source middleware for B2B and microservices-based environments.

Other vendors include Oracle, Red Hat, TIBCO Software and Scale Out Software.

How to choose a middleware platform?

Middleware should be chosen by considering what an individual or organization is looking to achieve with it. For example, if middleware is needed for data management, then database middleware should be used; if middleware is needed for application services, then application server middleware should be used. The key idea is to find the right software according to the performance needed.

Some middleware will offer specific tools to help developers. For example, Red Hat offers functions for container-based processing. If a specific tool set stands out, then that specific middleware may be worth it. However, organizations should be sure that the middleware will work with the applications it's trying to connect.

Organizations should also compare reliability, complexity and performance -- as some performance speeds may greatly vary -- with other middleware products.

What middleware can do?

· Execute any code.

· Make changes to the request and the response objects.

· End the request-response cycle.

· Call the next middleware in the stack.

Hint: If the current middleware function does not end the request-response cycle, it must call next() to pass control to the next middleware function. Otherwise, the request will be left hanging.

Types of express middleware

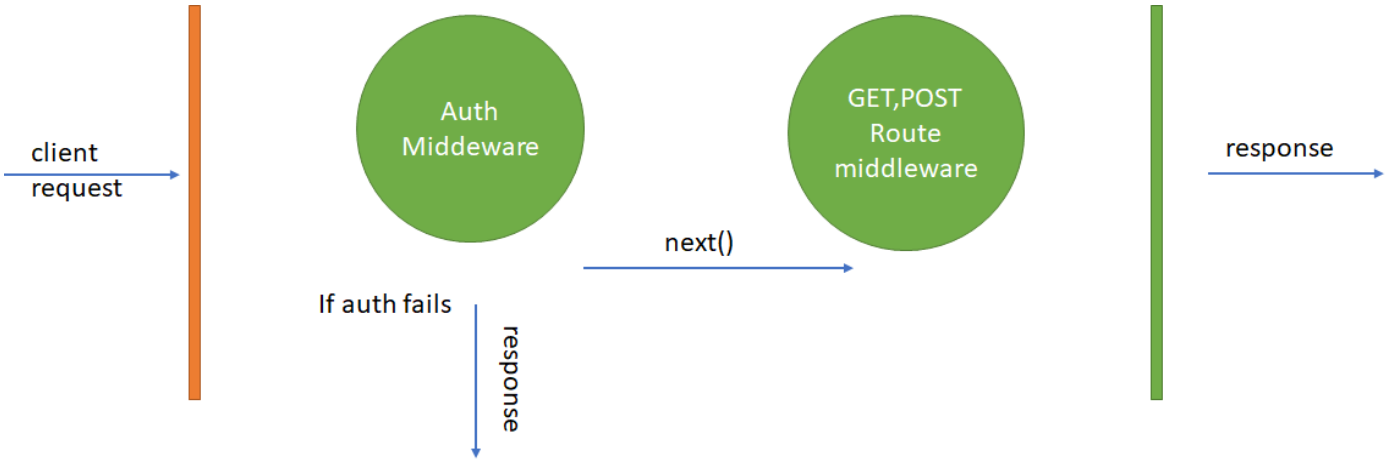
· Application-level middleware app.use

· Router level middleware router.use

- Built-in middleware express.static,express.json,express.urlencoded
- Error handling middleware app.use(err,req,res,next)
- Thirdparty middleware bodyparser,cookieparser

Example:

What does Auth middleware look like?



Using Express' Built-in Middleware

Built-in middleware functions are bundled with Express so we do not need to install any additional modules for using them.

Express provides the following Built-in middleware functions:

express.static	serves static assets

express.json	parses JSON payloads
express.urlencoded	parses URL-encoded payloads
express.raw	parses payloads into a Buffer and makes them available under req.body
express.text	parses payloads into a string

Resources:

Azure wikipedia techopedia turing medium reflectoring techtarget redhat