Express Middleware

[Burham Badr-Eldin Burham Soliman]

(1) What is Express Middleware ?

- We can literally define the middleware as the middle between a layer of the software and another, Which leads us to identify the Express Middleware as " the functions that's executes during the lifecycle of a request to the Express Server "
- By another meaning it can be the path or the route for the HTTP Request or Response to go through, and it can pass or terminate by the Middleware as per some predefined rules.

(2) Advantages of using middleware ?

- Middleware can process request objects multiple times before the server works for that request.
- Middleware can be used to add logging and authentication functionality.
- Middleware improves client-side rendering performance.
- Middleware is used for setting some specific HTTP headers.
- Middleware helps for Optimization and better performance.

(3) What is the meaning of Middleware Chaining ?

- Middleware can be chained from one to another, Hence creating a chain of functions that are executed in order. The last function sends the response back to the browser.
- Which mean that before sending the response back to the browser the different middlewares process the request.

(4) What is the disadvantages of Middleware ?

- Because of its prohibitively high development costs, not every business can afford to maintain and grow the potential of Middleware.
- Demands manual labor-intensive tasks which can affect the memory usage if they are not handled professionally.
- Provides no universal way of organizing things, which can be a problem for beginners and experienced developers, since in the absence of an organization this can lead to non-repairability of projects.

[*References*]

[1] Okta.com

[2] Geeks4Geeks.org

[3] Fourcornerstone.com

[4] cleveroad.com

[Shimaa Adel Mohammed Elzarief]

Middleware functions are functions that have access to the request object (`req`), the response object (`res`), and the next middleware function in the application's request-response cycle. The next middleware function is commonly denoted by a variable named `next`.

- ☐ 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.

Categories of Middleware in Express

- ☐ Application-level middleware:
  They represent middleware that are bound to the app object by using app.use and app.method().
  app.use will encompass all requests made, while app.method will be specific to the request type. (GET, POST, DELETE and PUT).

- ☐ Router-level middleware:
  Same as Application-level middleware, but the main difference is that it is bound to an instance of the Express router.

- ☐ Error handling middleware
  Works in the same way as other middleware functions, except that it has 4 arguments.

Using Express middlewares is simple. We can either pass them into app.use to make the middleware run for all request methods or app.METHOD to make them run for the given method.
We can call next to call the next middleware and call next('route') to call the route handler directly from a middleware.
Everything applies to both route and app-level middleware, except they bind to the express.Router() and express(), respectively.

[Mohamed Ahmed Hassan]

Express Middleware

What is middleware:-

Middleware is software that provides common services and capabilities to applications outside of what's offered by the operating system.

Data management, application services, messaging, authentication, and API management are all commonly handled by middleware.

Middleware helps developers build applications more efficiently. It acts like the connective tissue between applications, data, and users.

For organizations with multi-cloud and containerized environments, middleware can make it cost-effective to develop and run applications at scale.

The origin of middleware:-

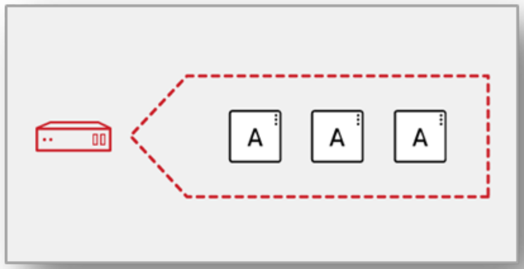
The term middleware first appeared in a report following the 1968 NATO Software Engineering conference in Garmisch-Partenkirchen, Germany. The conference sought to define the field of software engineering, and included software design, production, and distribution.

What kind of middleware are there:-

As a broad category, middleware can encompass everything from web servers to authentication systems to messaging tools. Here are a few of the common use cases for middleware in modern development.

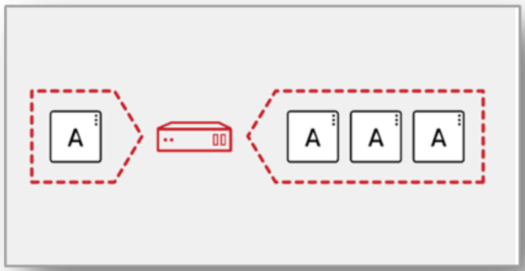
1-New Application development:-

Middleware can support modern and popular runtimes for a variety of use cases. Developers and architects can work with agility across platforms, following sets of foundational runtimes, frameworks, and programming languages. Middleware can also deliver commonly used functions such as web servers, single sign-on (SSO), messaging, and in-memory caching.



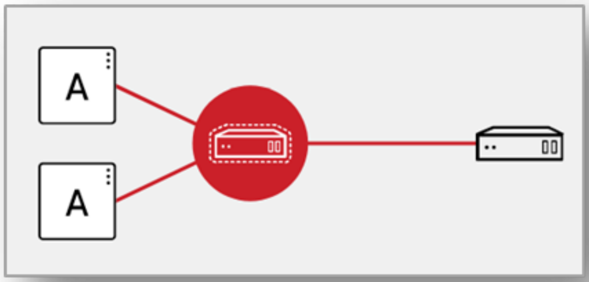
2-Optimization of existing applications:-

Middleware can help developers transform legacy monolithic applications into cloud-native applications, keeping valuable tools active with better performance and more portability.



<u>3-Comprehensive Integration:-</u>

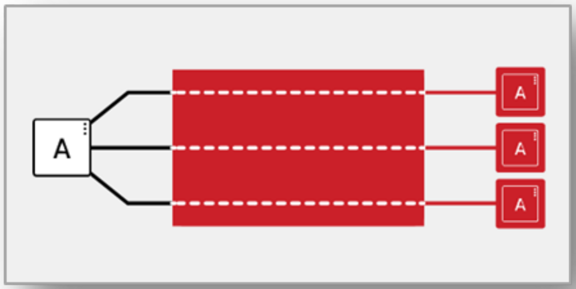
Middleware integration tools connect critical internal and external systems. Integration capabilities like transformation, connectivity, composability, and enterprise messaging, combined with SSO authentication, make it easier for developers to extend capabilities across different applications.



<u>4-Application Programming Interfaces (APIs):-</u>

Many middleware services are accessed through APIs, which are sets of tools, definitions, and protocols that allow applications to communicate with each other.

APIs make it possible to connect completely different products and services through a common layer.

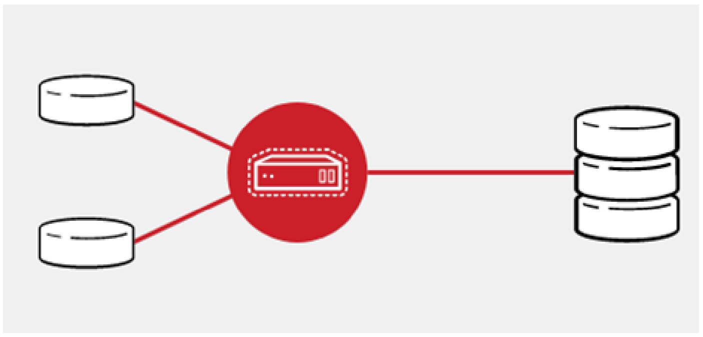


5-Intelligent business automation:-

Many middleware services are accessed through APIs, which are sets of tools, definitions, and protocols that allow applications to communicate with each other. APIs make it possible to connect completely different products and services through a common layer.

6-Data streaming:-

While APIs are one way to share data between applications, another approach is asynchronous data streaming. This replicates a data set in an intermediate store, where the data can be shared among multiple applications. One popular open source middleware tool for real-time data streaming is Apache Kafka.



Why is middleware important to cloud computing?

For all the benefits cloud-native development provides, it also brings added complexity. Applications can be deployed across multiple infrastructures, from on-premises systems to public clouds. Architectures can vary widely. Developers are juggling multiple tools, languages, and frameworks. And the pressure is on to do more in less time and at a lower cost.

Organizations turn to middleware as a way to manage this complexity and to keep application development quick and cost-effective. Middleware can support application environments that work smoothly and consistently across a highly distributed platform.

Build here. Deploy there. It works the same, thanks to the middleware beneath the applications.

What role does middleware play in app development?

Modern business apps are engineered to run at scale, on premises, and across clouds. To build them, developers need an application environment with unified foundational capabilities. Middleware is the key to assembling such an environment.

We can think of these capabilities in 4 layers, plus tooling:

1-The container layer:-

This layer of middleware manages the delivery aspect of application life-cycles in a uniform manner. It provides DevOps capability with CI/CD, container management, and service mesh capabilities.

2-The runtimes layer:-

This layer contains the execution environments for custom code. Middleware can provide lightweight runtimes and frameworks for highly distributed cloud environments such as micro services, in-memory caching for fast data access, and messaging for quick data transfer.

3-The integration layer:-

Integration middleware provides services to connect custom and purchased apps, as well as Software-as-a-Service (SaaS) assets through messaging, integration, and APIs to form functioning systems. It can also deliver in-memory database and data cache services, data/event streaming, and API management.

4-The process automation and decision management layer:-

This final layer of development middleware adds critical intelligence, optimization and automation, and decision management.

-Tooling:-

In addition to these 4 layers of middleware there's application development tooling. This allows teams to build applications using preset templates and containers, and facilitates efficient code sharing and joint development. Tooling supports a consistent and coherent application development and delivery experience on-premises and cloud.

Sources:

[What is middleware? (redhat.com)]

[Middleware - Wikipedia]

[What is Middleware? | IBM]

[What is Middleware? (And How Does it Work?) | Talend]

[What is Middleware? - Middleware Explained - AWS (amazon.com)]

By: Mohamed Hassan, Advanced Track Web development UDACITY

To: Mr. HOSSAM ABUBAKR