

Middleware function<sup>1</sup> - This is a function/program that runs between the time a server gets a request from a client and the time the server sends the response out to the client (they execute during the lifecycle of a request to the Express server.). These functions have access to the request object (req), the response object (res), and the next middleware function in the application's request-response cycle.

Middleware functions can: Execute any code, Make changes to the request and the response objects, End the request-response cycle, Call the next middleware function in the stack.

## Types of middleware

**Application-level middleware:** This middleware function is bound to the instance of the app object by using app. use() or app. METHOD functions (where method is the HTTP method of the request).

**Router-level middleware:** Works in the same way as application-level middleware, except it is bound to an instance of express.Router().

**Error-handling middleware:** always takes four arguments (err, req, res, next). Even when the next object is not used, it is passed as an argument to this middleware as a signature, otherwise it will be interpreted as regular middleware and will fail to handle errors.

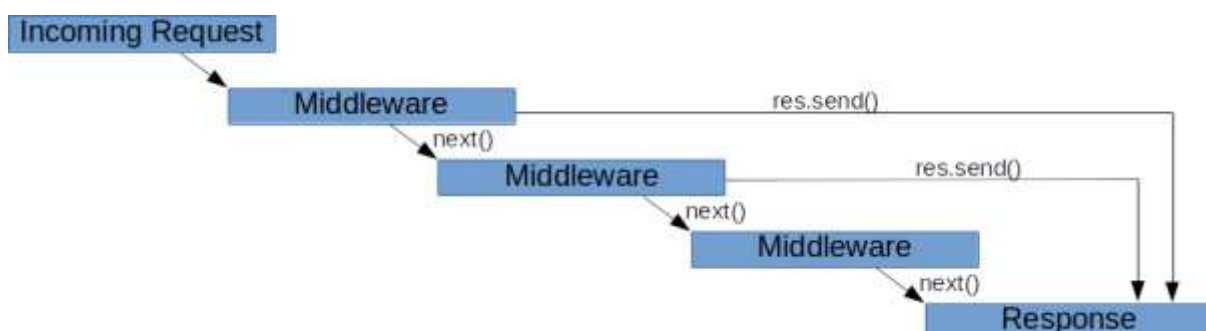
**Built-in middleware:** these come inbuilt with the express framework and can be called just like other express modules.

Express' built-in middleware functions:

- express.static: serves static assets such as HTML files, images, and so on.
- express.json: parses incoming requests with JSON payloads. NOTE: Available with Express 4.16.0+
- express.urlencoded: parses incoming requests with URL-encoded payloads. NOTE: Available with Express 4.16.0+

**Third-party middleware:** these are already build middleware modules that can be installed in our express app using the npm install command and then required in the express app file. They add functionality to Express apps e.g. bodyParser, cookie-parser e.t.c.

## Middleware Order<sup>2</sup>



<sup>1</sup> <https://expressjs.com/en/guide/using-middleware.html>

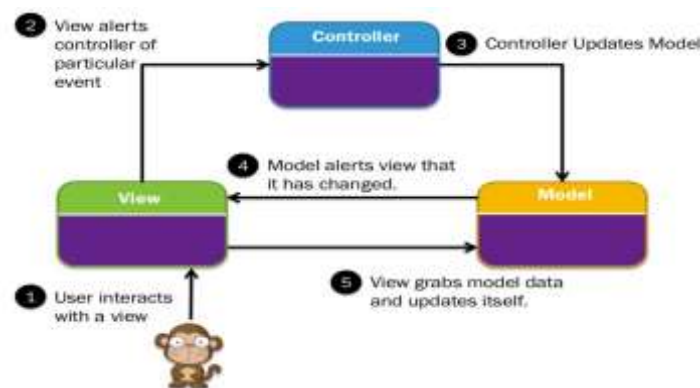
<sup>2</sup> <https://developer.okta.com/blog/2018/09/13/build-and-understand-express-middleware-through-examples>

## MVC Structure<sup>3</sup>

**Model-View-Controller (MVC)** framework is an architectural pattern that separates an application into three main logical components Model, View, and Controller/router.

**History:** MVC architecture first discussed in 1979 by Trygve Reenskaug. MVC model was first introduced in 1987 in the Smalltalk programming language. The pattern was first accepted as a general concept in a 1988 article and widely used today in modern web applications.

### MVC Architecture



- Model: It includes all the data and its related logic
- View: Present data to the user or handles user interaction
- Controller: An interface between Model and View components

### Advantages of MVC:

- Easy code maintenance easy to extend and grow
- MVC Model component can be tested separately from the user hence effective for TDD.
- Easier support for new type of clients
- Development of the various components can be performed parallelly.
- Avoids complexity by dividing an application into the three units. Model, view, and controller
- Only uses a Front Controller pattern which process web application requests through a single controller.
- Works well for Web apps which are supported by large teams of web designers and developers.
- Provides clean separation of concerns (SoC).
- Search Engine Optimization (SEO) Friendly.
- MVC allows logical grouping of related actions on a controller together.

### Disadvantages of using MVC

- Difficult to read, change, to unit test, and reuse this model.
- New layers of abstraction can sometimes make it complex.
- No formal validation support.
- Increased complexity and Inefficiency of data
- The difficulty of using MVC with the modern user interface
- There is a need for multiple programmers to conduct parallel programming.
- Knowledge of multiple technologies is required.
- Maintenance of lots of codes in Controller.

<sup>3</sup> <https://www.guru99.com/mvc-tutorial.html>