



Express JS

Aruna S

Department of
Computer Science and Engineering

EXPRESS JS

Express Middlewares

S. Aruna

Department of Computer Science and Engineering

In express, middleware functions are the functions which have access to the request and response objects for tasks like parsing request bodies, adding response headers, etc. along with the next function present in the application's request-response cycle.

These functions are capable of performing the below-listed tasks:

- Execution of any code
- Modify the request and the response objects.
- End applications request-response cycle.
- Call the next middleware present in the cycle.

In case, the current function doesn't terminate the request-response cycle then it must invoke the `next()` function in order to pass on the control to the next available middleware function.

If not done, then the request will be left incomplete.

Below is the list of majorly used middleware in any Express.js application:

- Application-level middleware
- Router-level middleware
- Error-handling middleware
- Built-in middleware
- Third-party middleware

```
var express = require('express');
var app = express();
//MIDDLEWARE FUNCTION TO LOG REQUEST PROTOCOL
app.use('/', function(req, res, next){
    console.log("A request for things received at " + Date.now())
});
    next();
});
// ROUTE HANDLER THAT SENDS THE RESPONSE
app.get('/', function(req, res){
    res.send('Things');
});
app.listen(3000);
```

<http://expressjs.com/en/resources/middleware.html>

some of the most commonly used middleware

body-parser

This is used to parse the body of requests which have payloads attached to them. To mount body parser, we need to install it using

npm install body-parser

and to mount it, include the following lines (Parse incoming request bodies in a middleware before your handlers, available under the req.body property)

```
var bodyParser = require('body-parser');  
//To parse URL encoded data  
app.use(bodyParser.urlencoded({ extended: false }));  
//To parse json data  
app.use(bodyParser.json());
```

cookie-parser

It parses Cookie header and populate req.cookies with an object keyed by cookie names. To mount cookie parser, we need to install it using npm install cookie-parser and to mount it, include the following lines.

```
var cookieParser = require('cookie-parser');  
app.use(cookieParser());
```



THANK YOU

Aruna S

Department of
Computer Science and Engineering

arunas@pes.edu