Express.js- web framework based-Node.js http module.

app.use([path,] callback[, callback...])

Middleware has 4 distinct methods: express.json([options],urlencoded([options]),raw([options]),text([options])

NOTE: All built-in middleware are based on body-parser module. It does not support multipart( )

Using body parsing Only for certain route

**require(**'express'**);** express**();const** jsonParser **=** express**.**json**();**

**const** urlencodedParser **=** express**.**urlencoded**({** extended**: false });**

app**.**use**(**'/login'**,** urlencodedParser**, function (**req**,** res**) {});**

app.use(express.static(path.join(\_\_dirname, 'public'))); <link rel="stylesheet" href="/css/main.css">

res.status(404).sendFile(path.join(\_\_dirname, 'views', '404.html'))

Middleware Order Matters

Controller

require('../models/product'); exports.getProducts

Route

require('express'); require('../controllers/productController'); const router = express.Router(); router.get('/', productController.getProducts); module.exports = router;

app.js

require('express'); require('./routes/product'); require('cors'); const app = express(); app.use(cors()); app.use(express.json()); app.use('/products', productRouter);

app.use((req, res, next) => {res.status(404).json({ error: “”}); app.use((err, req, res, next) => {res.status(500).json({ error: “” });app.listen(3000,()=>

**JWT** in controller

**Compact**: smaller size-transmission is fast.

**Self-contained**: The payload contains all information about the user,

**Claims** are statements about an entity (typically, the user) and additional metadata. There are three types of claims:

const jwt = require('jsonwebtoken'); accessTokenSecret ,

exports.login = (req, res, next) => {

const user = new User(req.body.username, req.body.password, null).login(); if (user) {

const accessToken = jwt.sign({ username: user.username, role: user.role }, accessTokenSecret); res.json({ accessToken });

} else {

res.status(200).json({ 'error': 'username or password invalid' });

}

}

exports.authorize = (req, res, next) => {

const authHeader = req.headers.authorization;

if (authHeader) {

const token = authHeader.split(' ')[1];

jwt.verify(token, accessTokenSecret, (err, user) => {

if (err) {

return res.status(403).json({ "error": "Forbidden" });

}

req.user = user;

next();

});

} else {

res.status(401).json({ "error": "Unauthorized" });

Route

const express = require('express');

const authController = require('../controllers/authController'); const router = express.Router();

router.post('/login', authController.login); router.use(authController.authorize);

module.exports = router;