Simple RESTApi NodeJS ft MongoDB

Joshua Permana

What is RESTAPI?

Application Programming Interface (API) has become an part of software development. REST is an acronym for Representational State Transfer, In a RESTful Web service, requests made to a resource's URI will elicit a response with a payload formatted in HTML, XML, JSON, or some other format.

Why using RESTAPI?

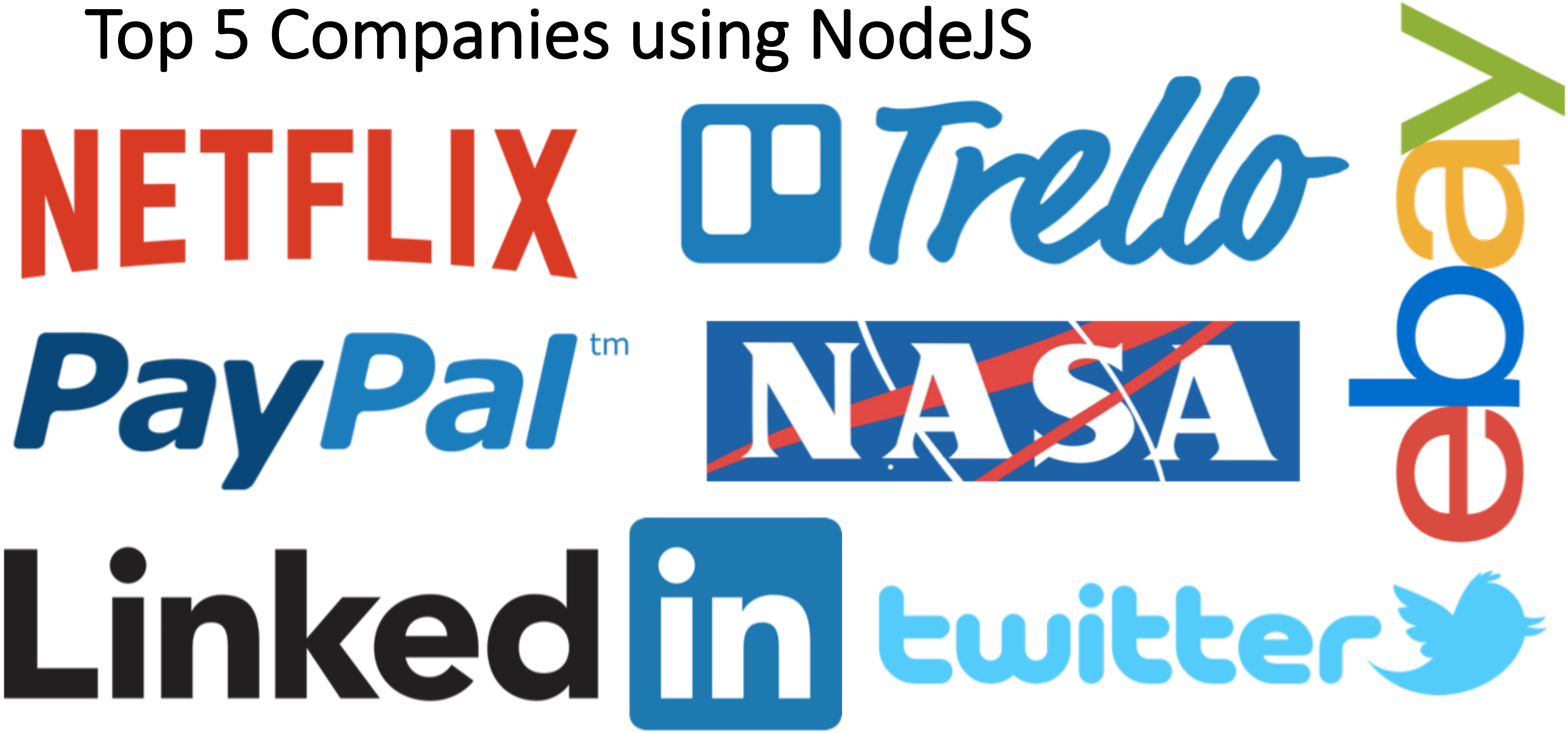
“RESTful systems aim for fast performance, reliability, and the ability to grow, by re-using components that can be managed and updated without affecting the system as a whole, even while it is running.”

-Wikipedia

What is NodeJS?

“NodeJS came into existence when the original developers of JavaScript extended it from something you could only run in the browser to something you could run on your machine as a standalone application.” –freecodecamp

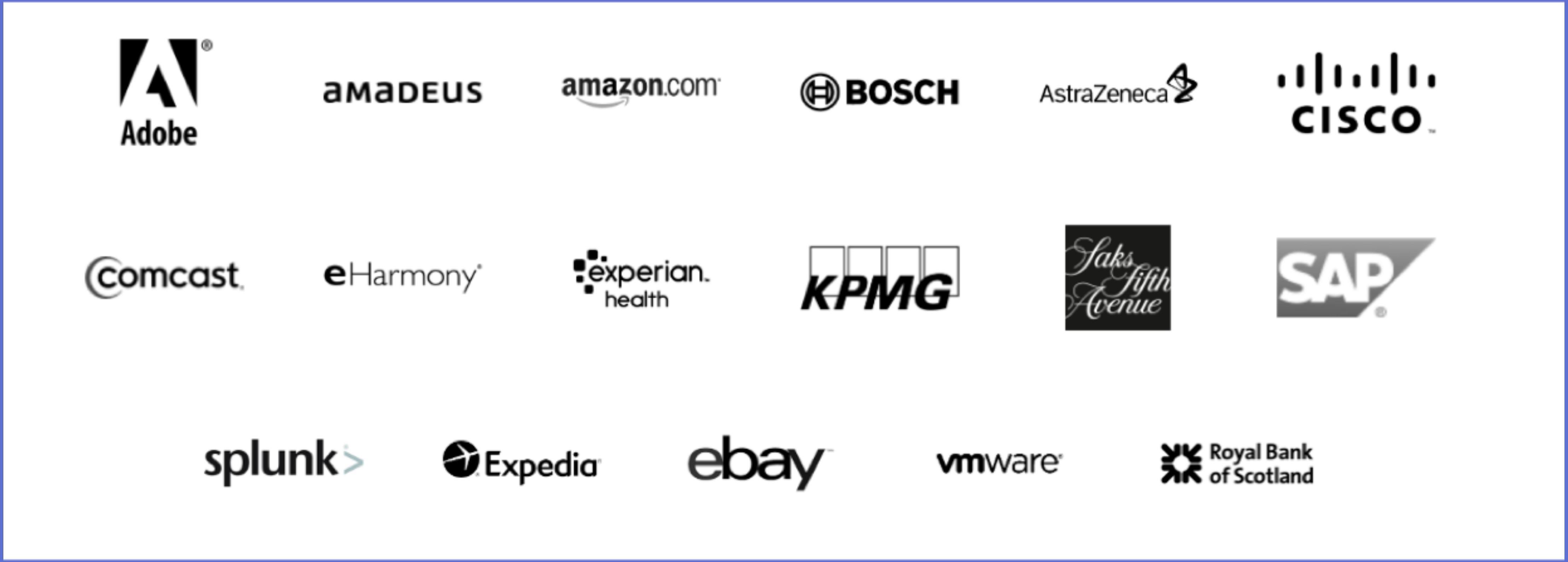
Top 5 Companies using NodeJS



What is MongoDB?

MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need

Top 10 Companies using MongoDB



Set Environment

Just make sure everything is okay

- Text Editor (Vim, Atom, Sublime, VSCODE, etc)
- HTTP client for testing web services (Postman, Insomnia, etc)
- Node (nodejs.org/en/download)
- NPM (npm -v)
- Mongo (

windows : https://s.id/4VSXb

mac : brew update -> brew install mongodb ->
mkdir -p /data/db ->
sudo chown -R `id -un` /data/db

linux : https://s.id/4VT2u

)

MongoDB basic steps

Create :

```
db.(namadb).insert({nama:”KedaiDesa”, alamat:”TamKul”})
```

Read :

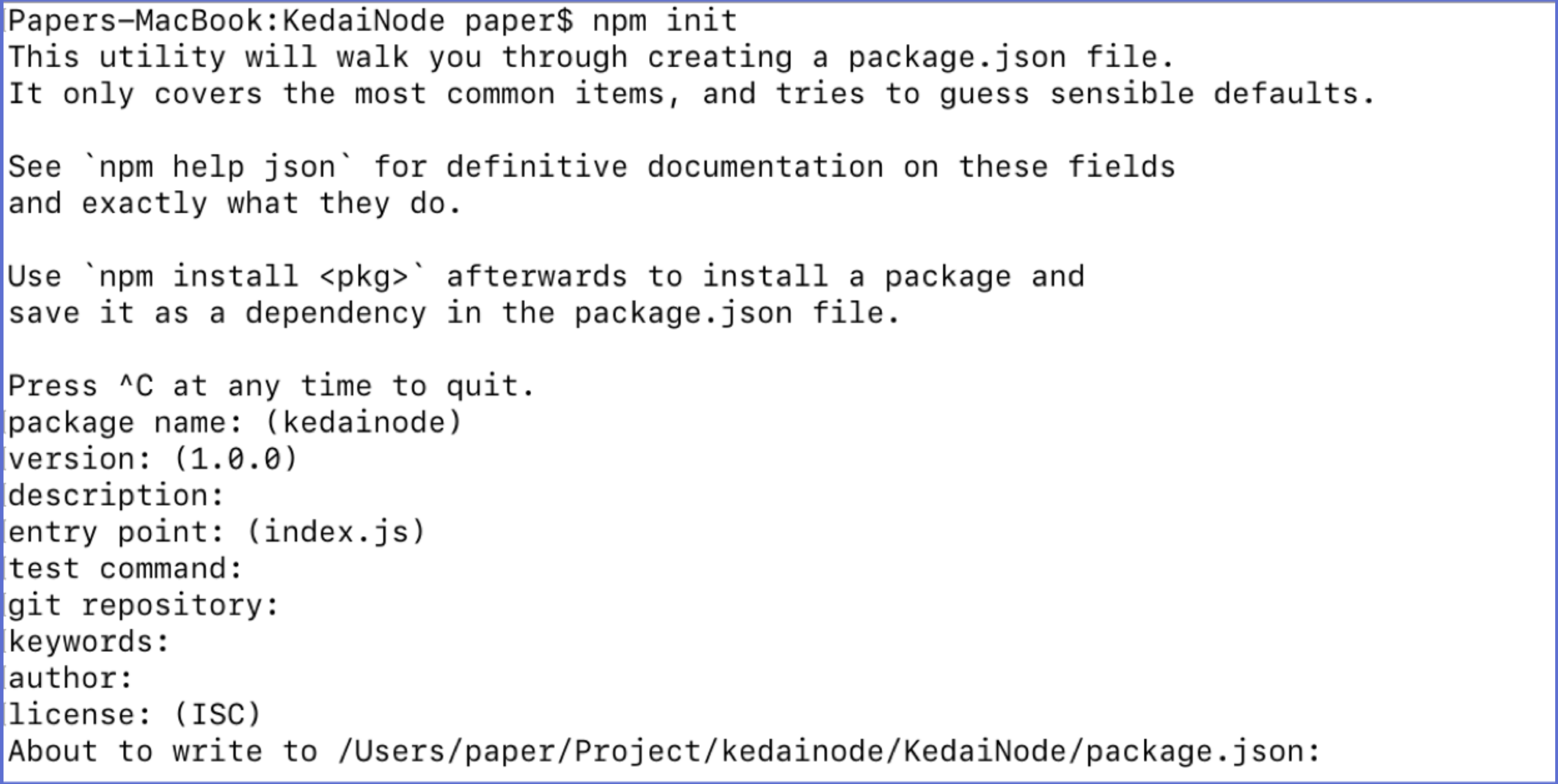
```
db.(namadb).find()
```

Update :

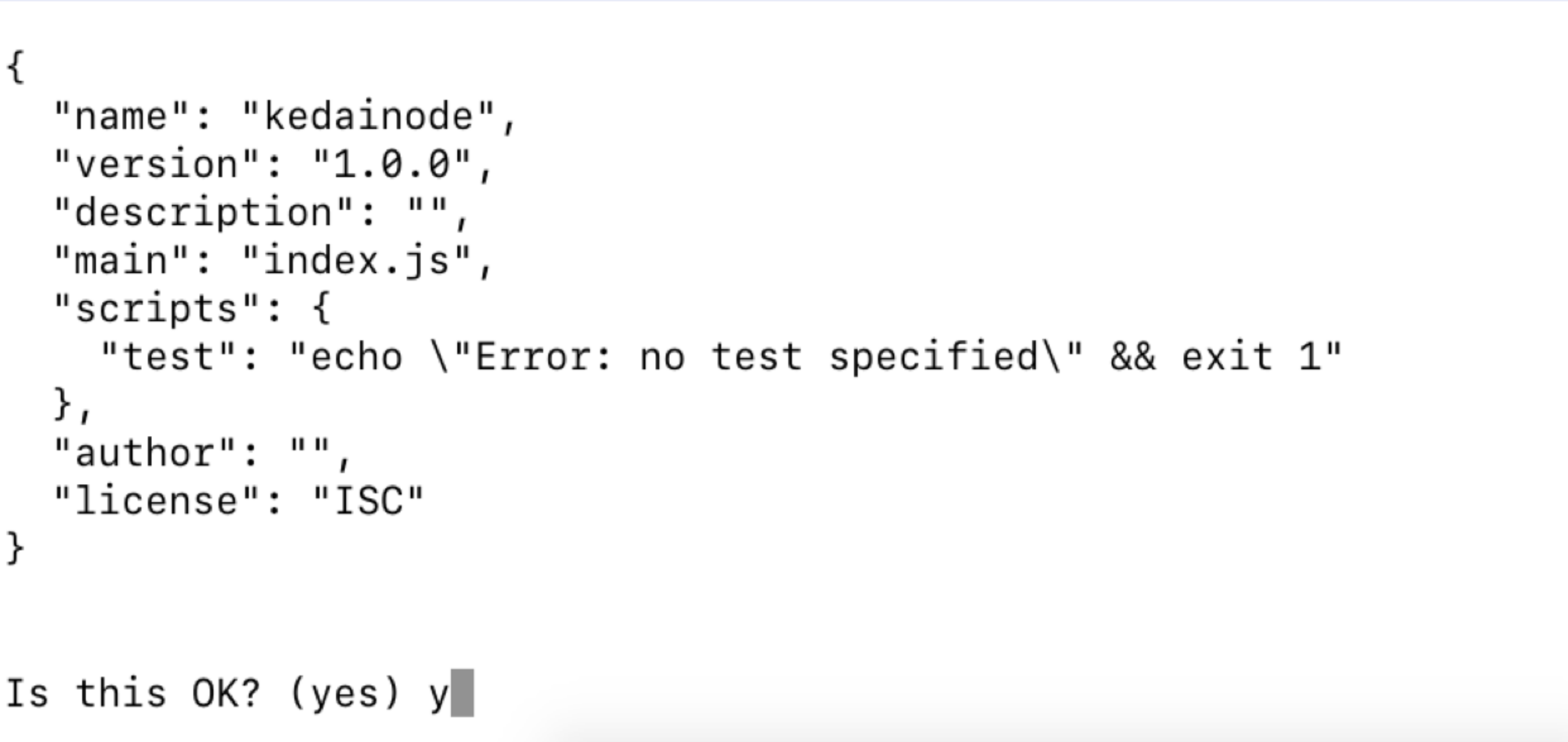
```
db.(namadb).update({nama:”KedaiDesa"}, {$set:{nama:”Kedainode"}})
```

Let’s get started

1. Create project folder using terminal command “mkdir KedaiNode”
2. Change your current location into project folder “cd KedaiNode”
3. Run ”npm init”

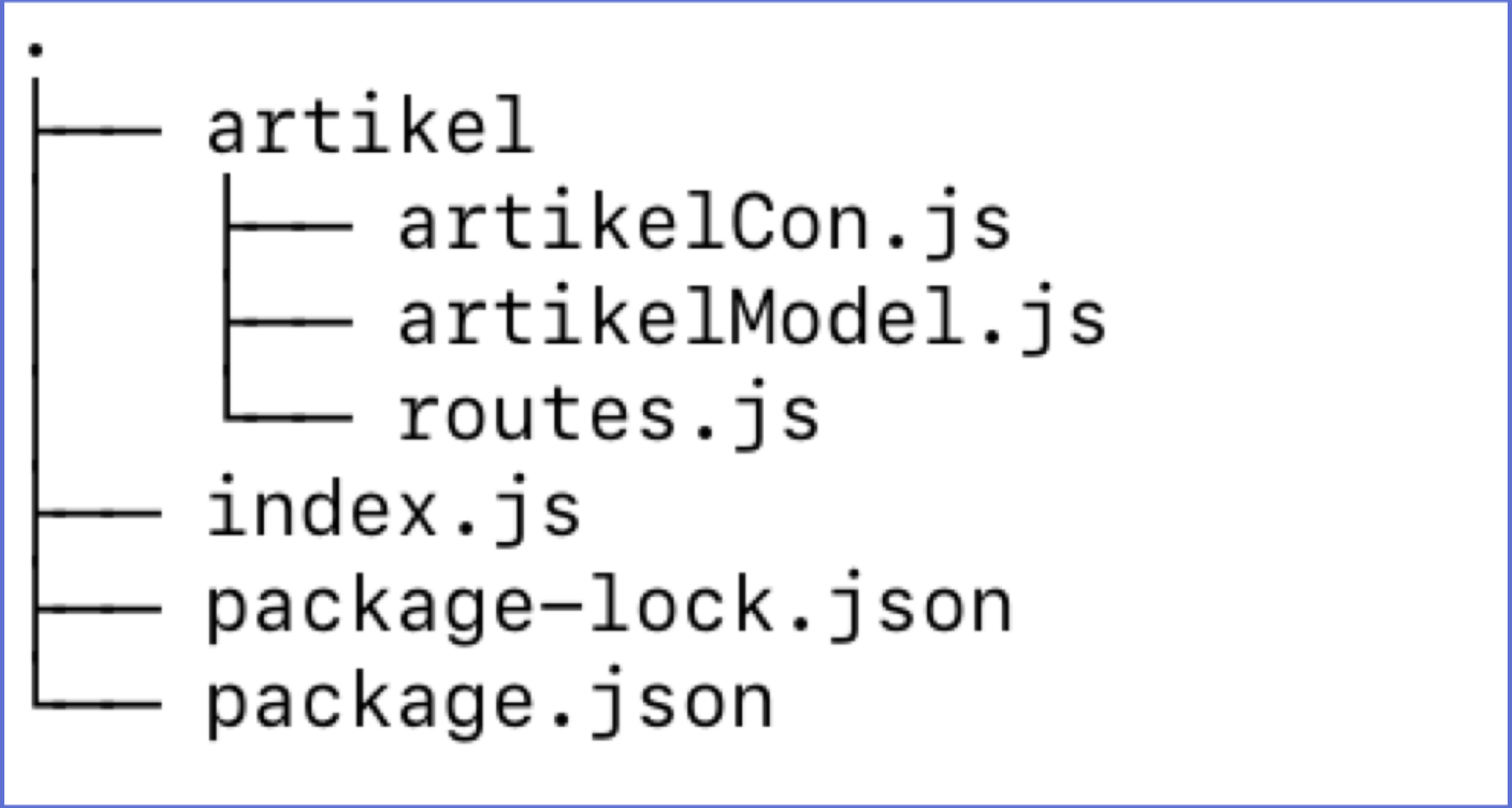


```
Papers-MacBook:KedaiNode paper$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (kedainode)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /Users/paper/Project/kedainode/KedaiNode/package.json:
```

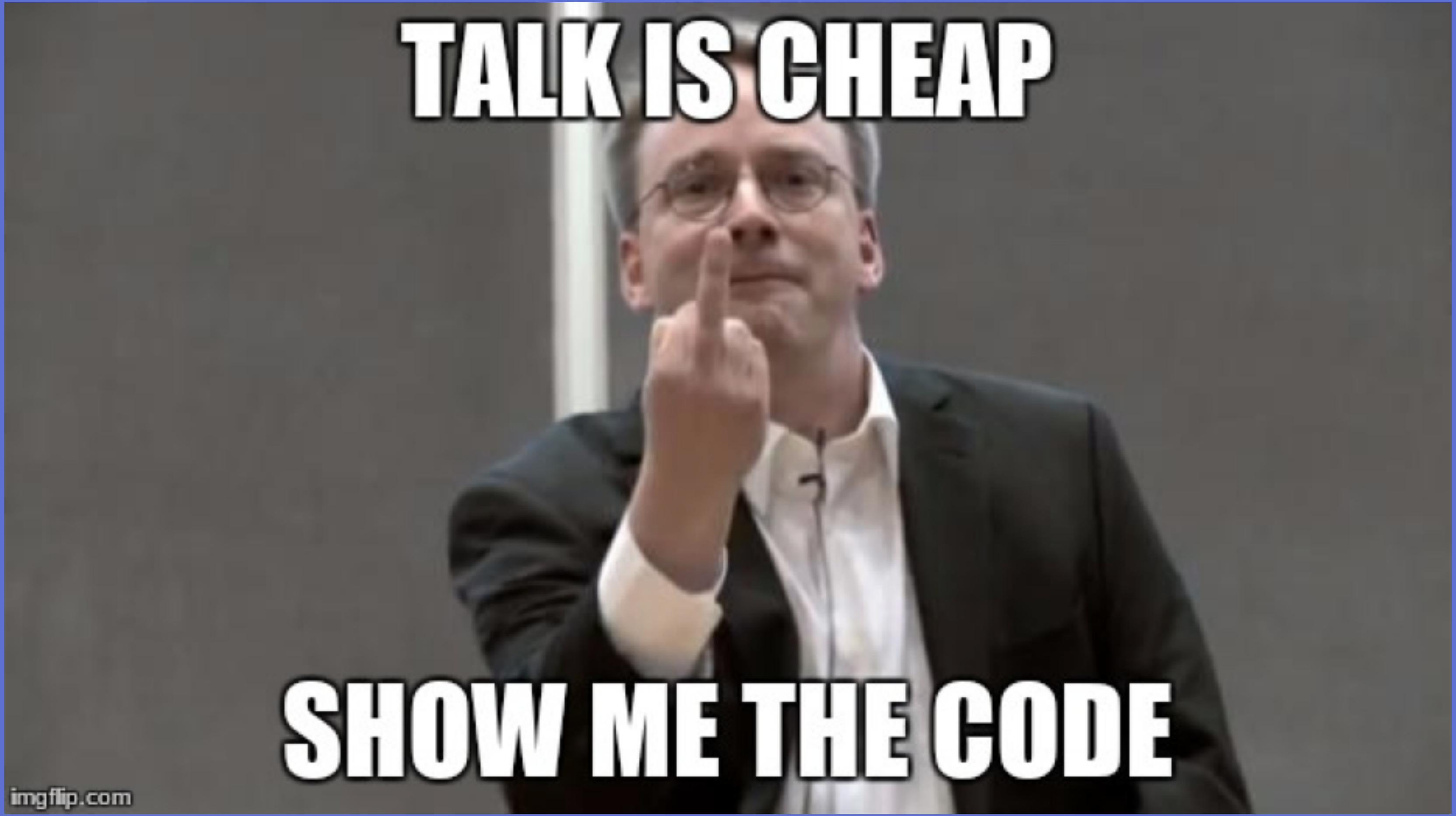


```
{
  "name": "kedainode",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this OK? (yes) y
```

Project Structure

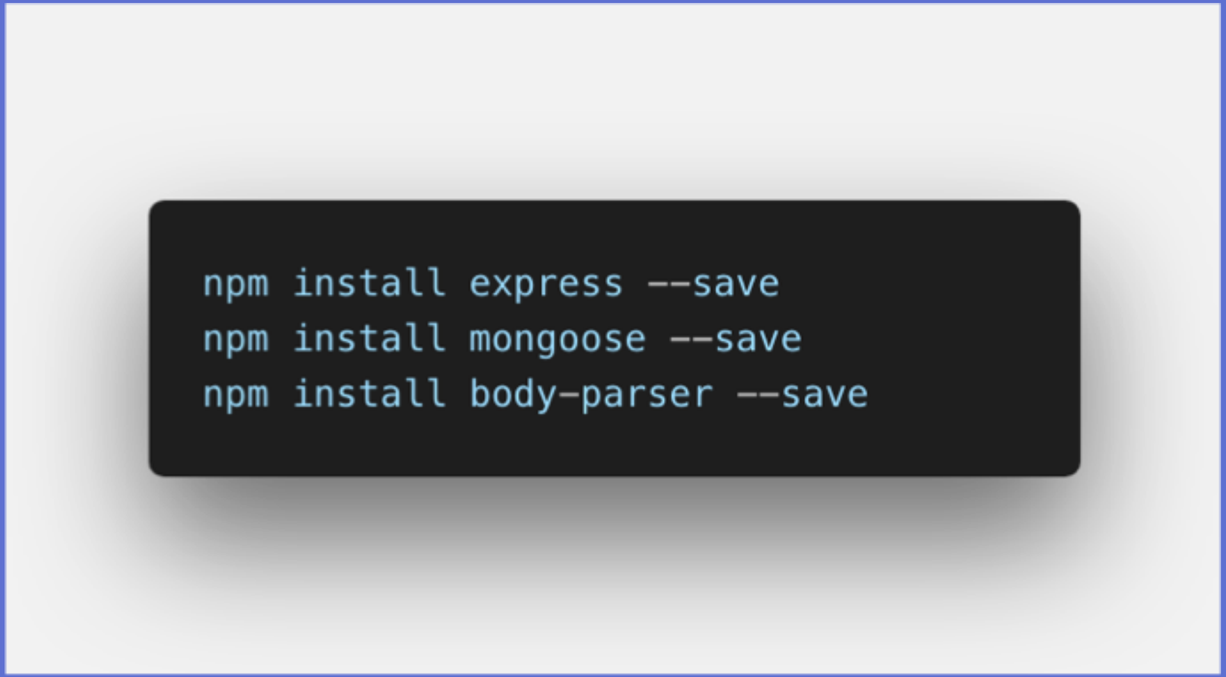


```
.
├── artikel
│   ├── artikelCon.js
│   ├── artikelModel.js
│   └── routes.js
├── index.js
├── package-lock.json
└── package.json
```


TALK IS CHEAP
SHOW ME THE CODE
imgflip.com

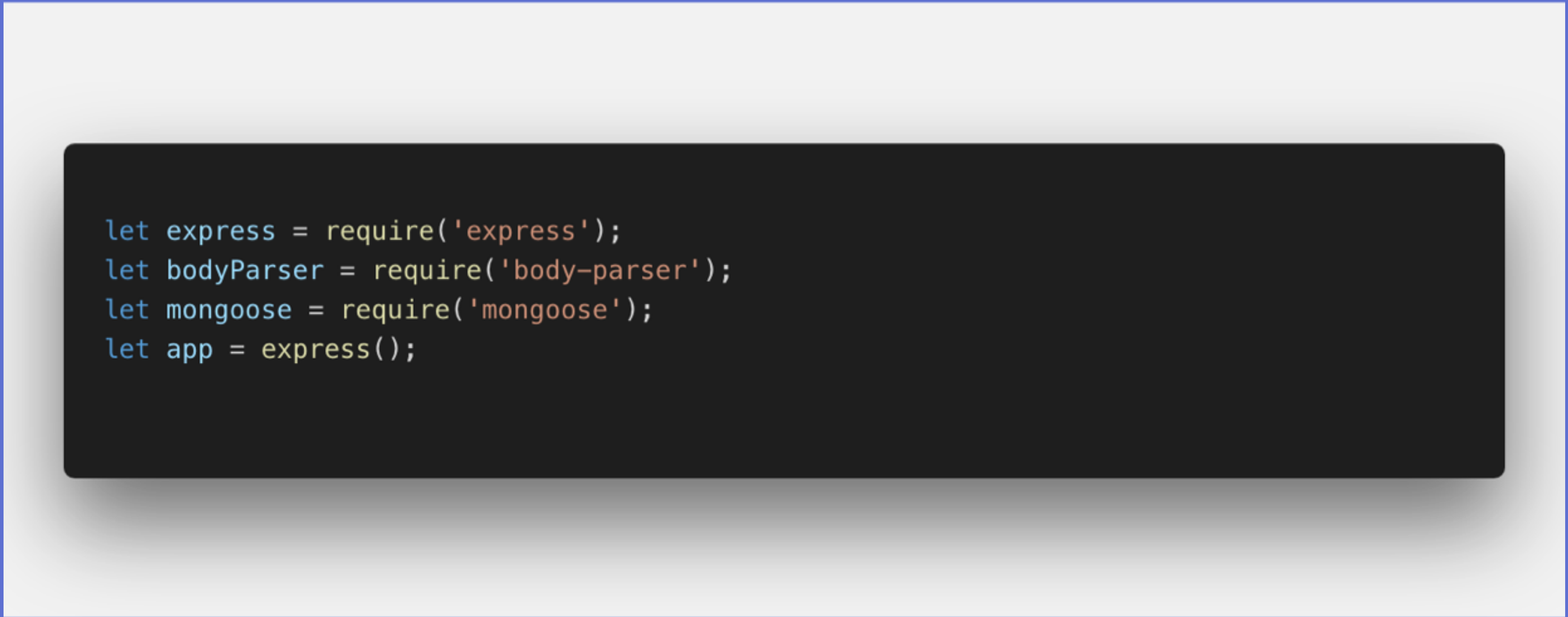
Library used in this project

- Open terminal and go to project folder run :



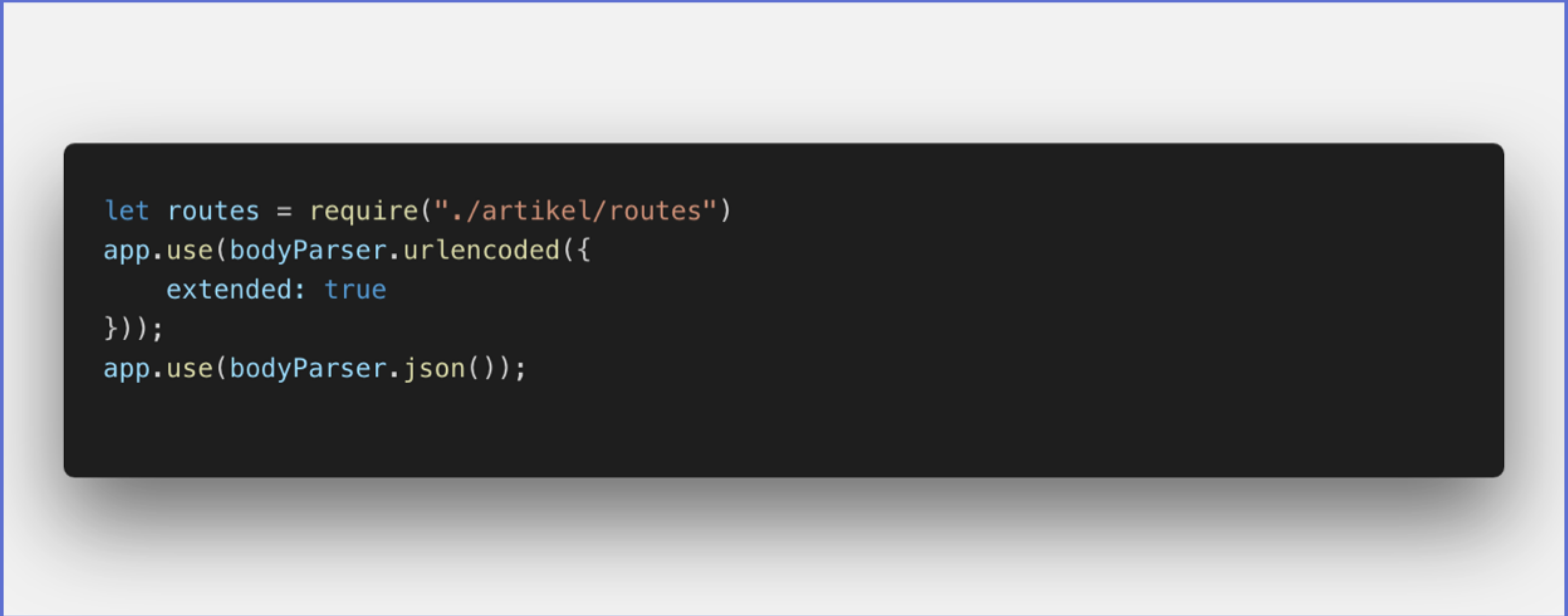
```
npm install express --save
npm install mongoose --save
npm install body-parser --save
```

Index.js



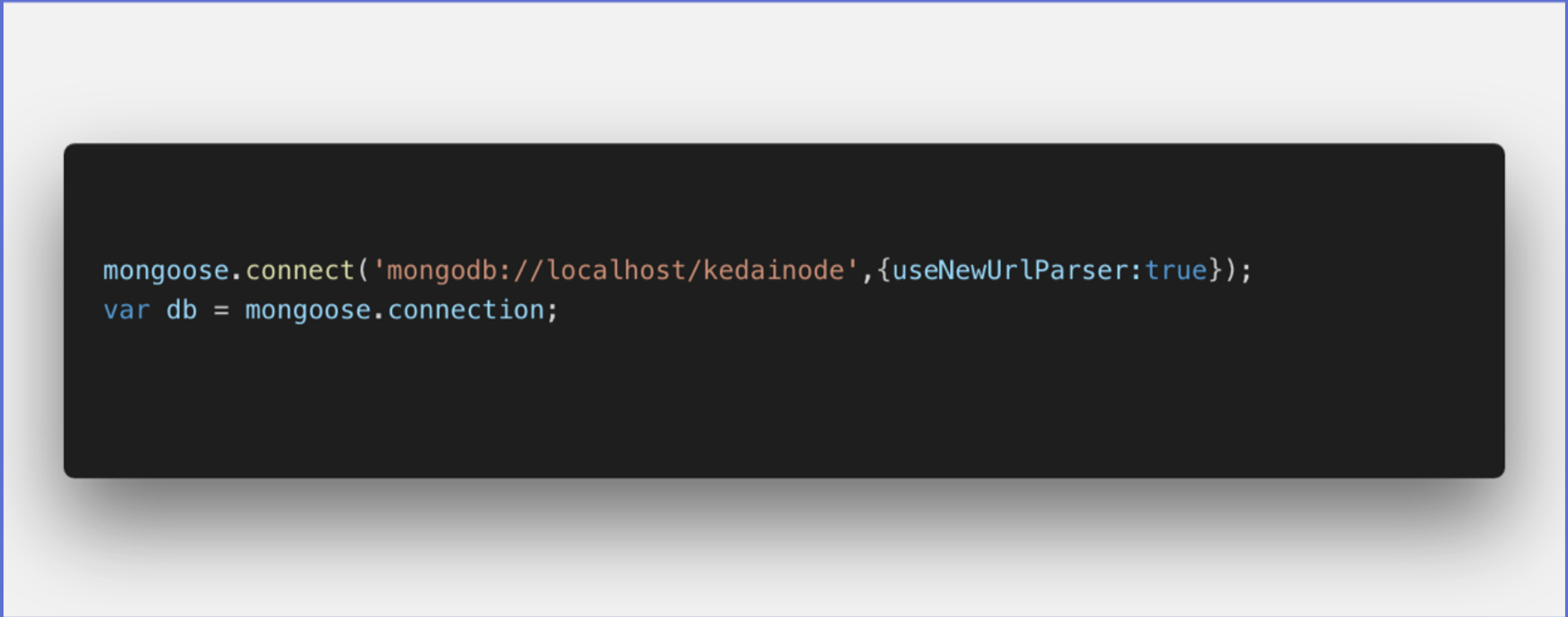
```
let express = require('express');
let bodyParser = require('body-parser');
let mongoose = require('mongoose');
let app = express();
```

Index.js



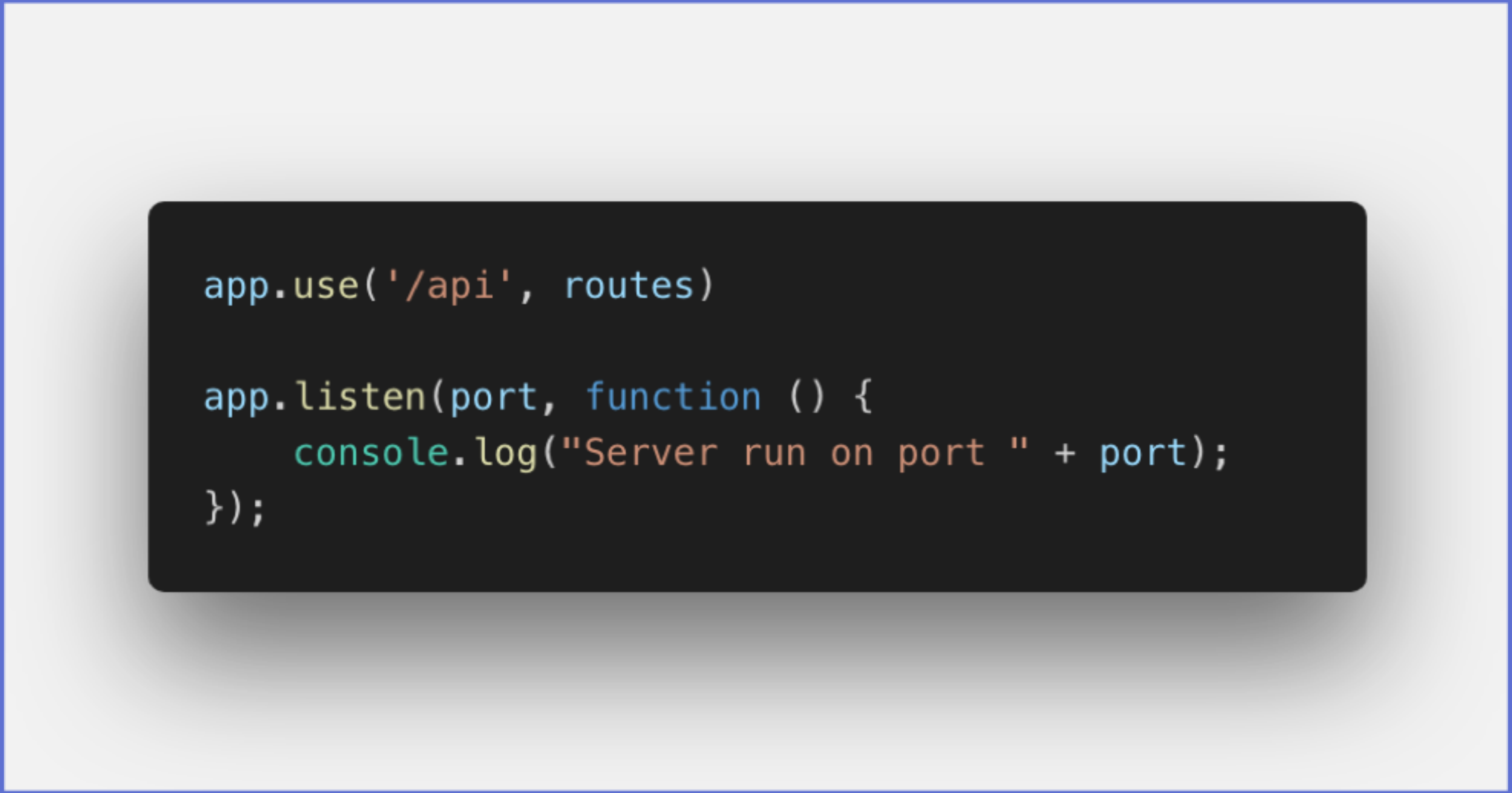
```
let routes = require("./artikel/routes")
app.use(bodyParser.urlencoded({
    extended: true
}));
app.use(bodyParser.json());
```

Index.js



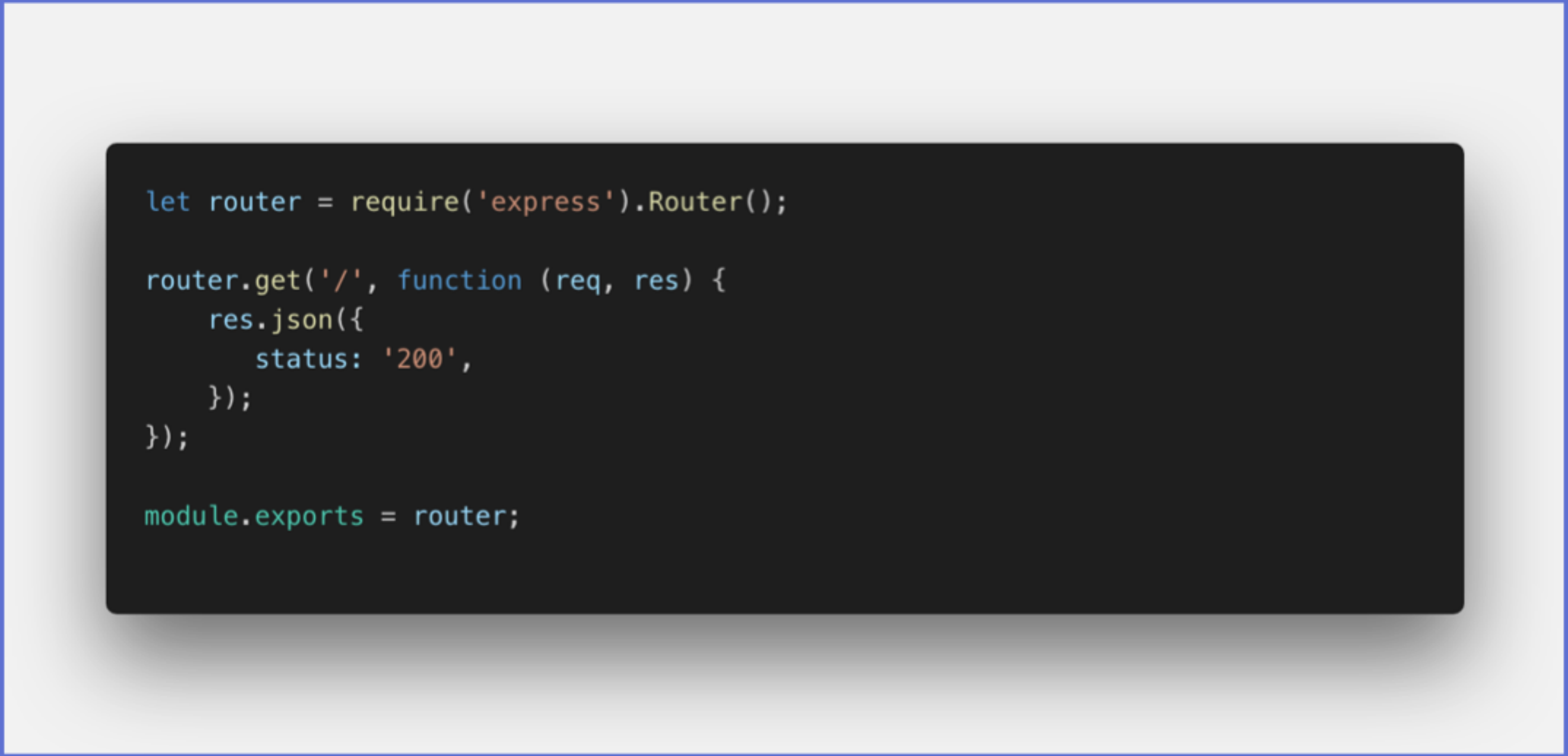
```
mongoose.connect('mongodb://localhost/kedainode',{useNewUrlParser:true});
var db = mongoose.connection;
```

Index.js



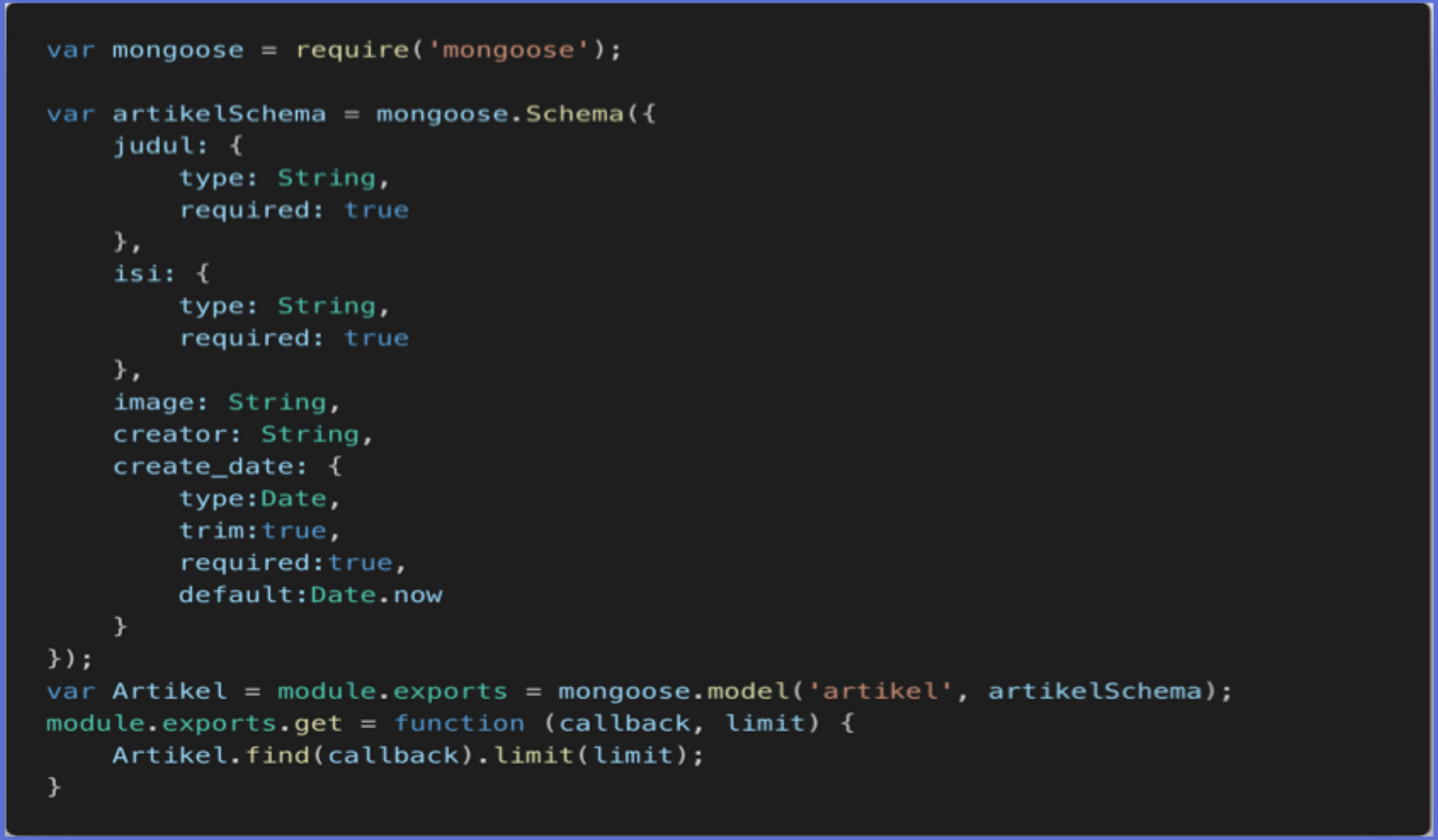
```
app.use('/api', routes)

app.listen(port, function () {
    console.log("Server run on port " + port);
});
```

Artikel/routes.js



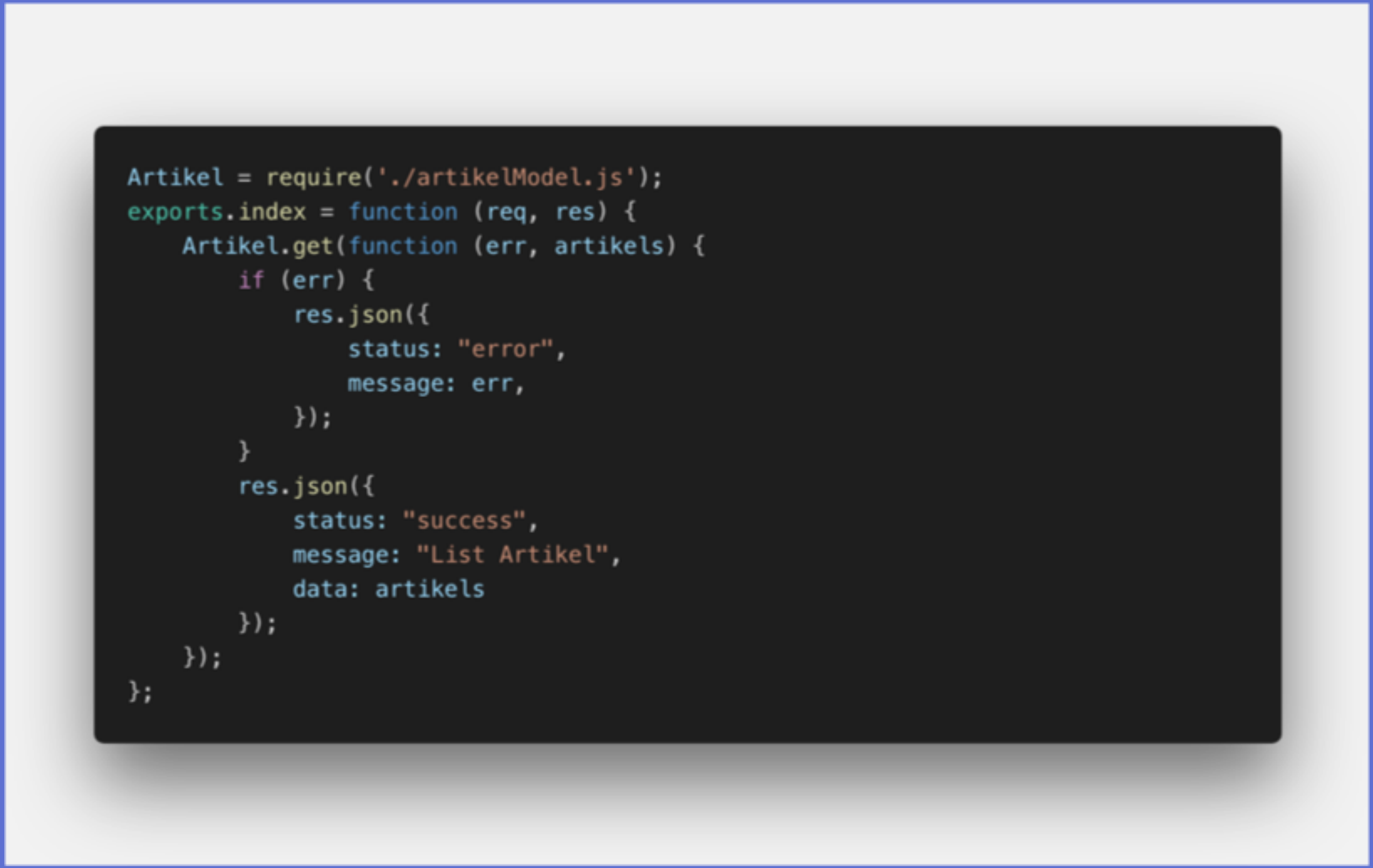
```
let router = require('express').Router();

router.get('/', function (req, res) {
    res.json({
        status: '200',
    });
});

module.exports = router;
```

Artikel/artikelModel.js



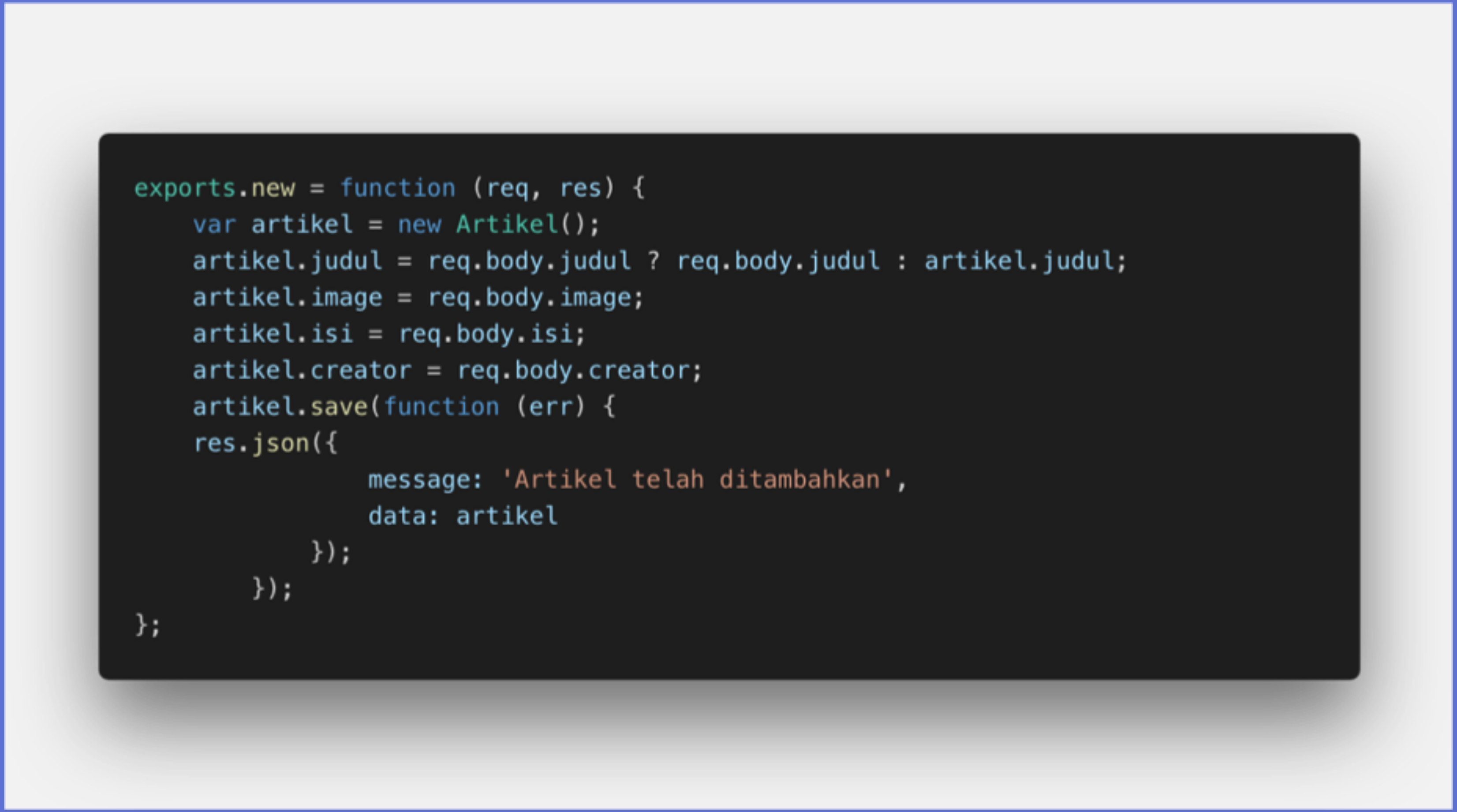
```
var mongoose = require('mongoose');

var artikelSchema = mongoose.Schema({
    judul: {
        type: String,
        required: true
    },
    isi: {
        type: String,
        required: true
    },
    image: String,
    creator: String,
    create_date: {
        type:Date,
        trim:true,
        required:true,
        default:Date.now
    }
});
var Artikel = module.exports = mongoose.model('artikel', artikelSchema);
module.exports.get = function (callback, limit) {
    Artikel.find(callback).limit(limit);
}
```

Artikel/artikelCon.js



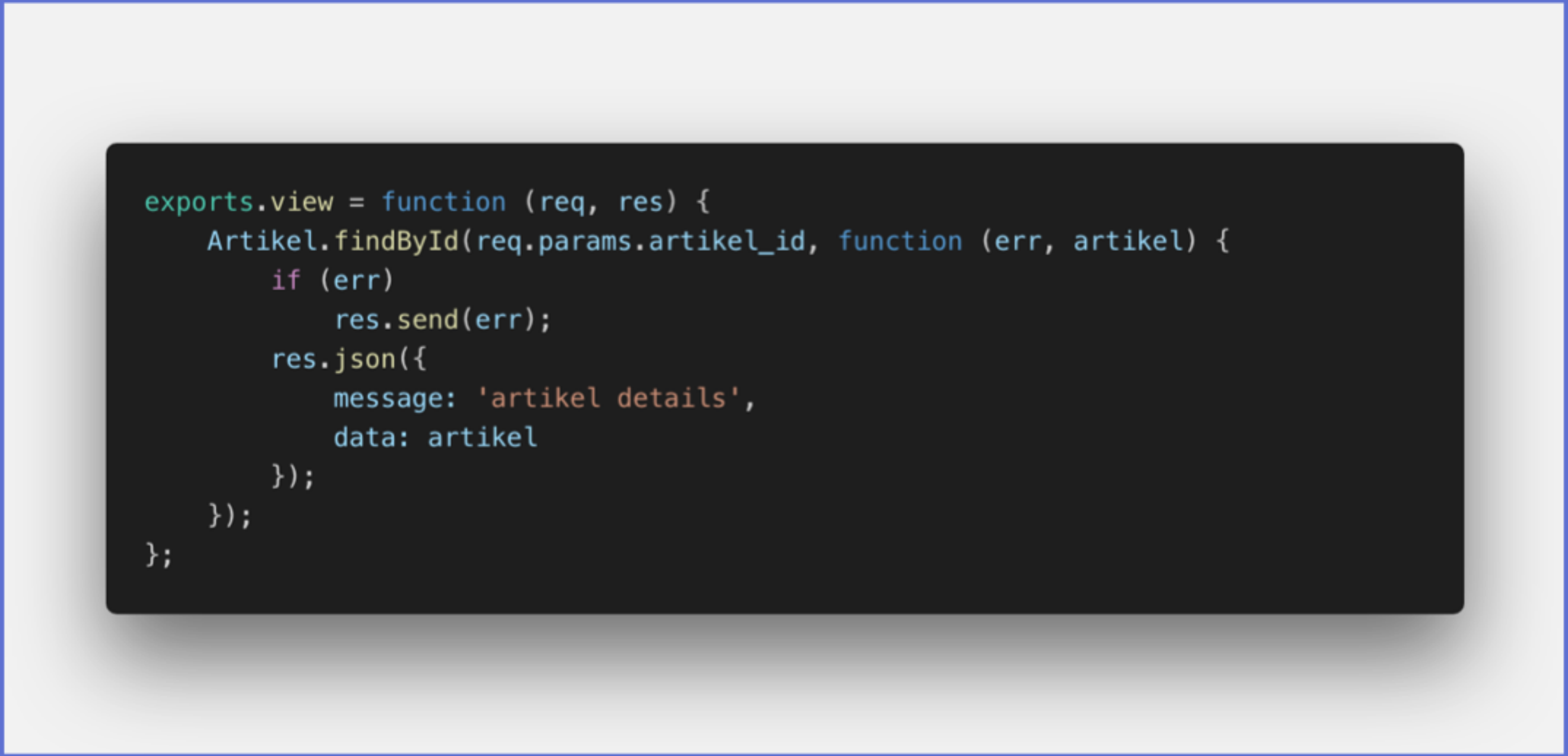
```
Artikel = require('./artikelModel.js');
exports.index = function (req, res) {
    Artikel.get(function (err, artikels) {
        if (err) {
            res.json({
                status: "error",
                message: err,
            });
        }
        res.json({
            status: "success",
            message: "List Artikel",
            data: artikels
        });
    });
};
```

Artikel/artikelCon.js



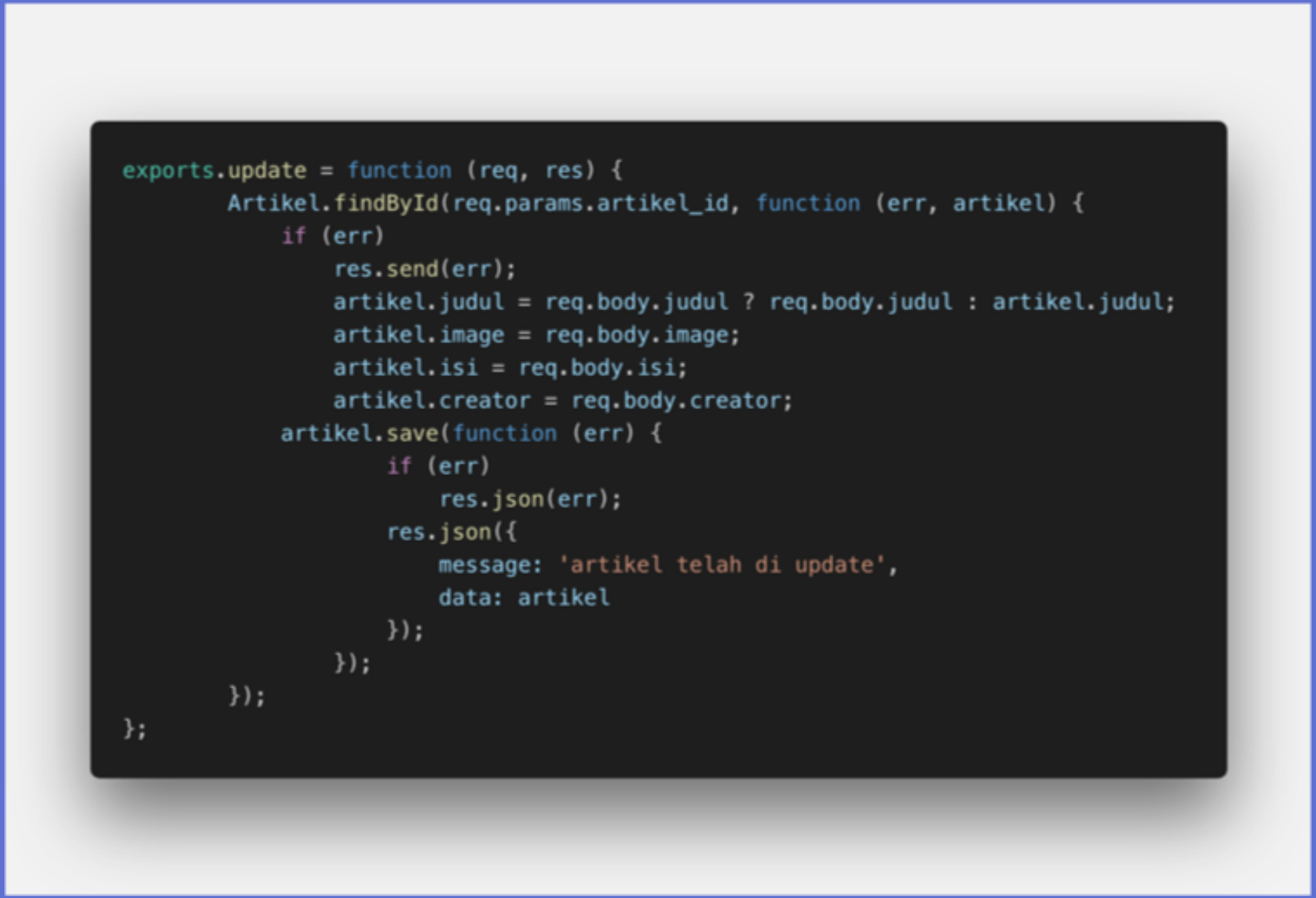
```
exports.new = function (req, res) {
    var artikel = new Artikel();
    artikel.judul = req.body.judul ? req.body.judul : artikel.judul;
    artikel.image = req.body.image;
    artikel.isi = req.body.isi;
    artikel.creator = req.body.creator;
    artikel.save(function (err) {
    res.json({
                message: 'Artikel telah ditambahkan',
                data: artikel
            });
        });
};
```

Artikel/artikelCon.js



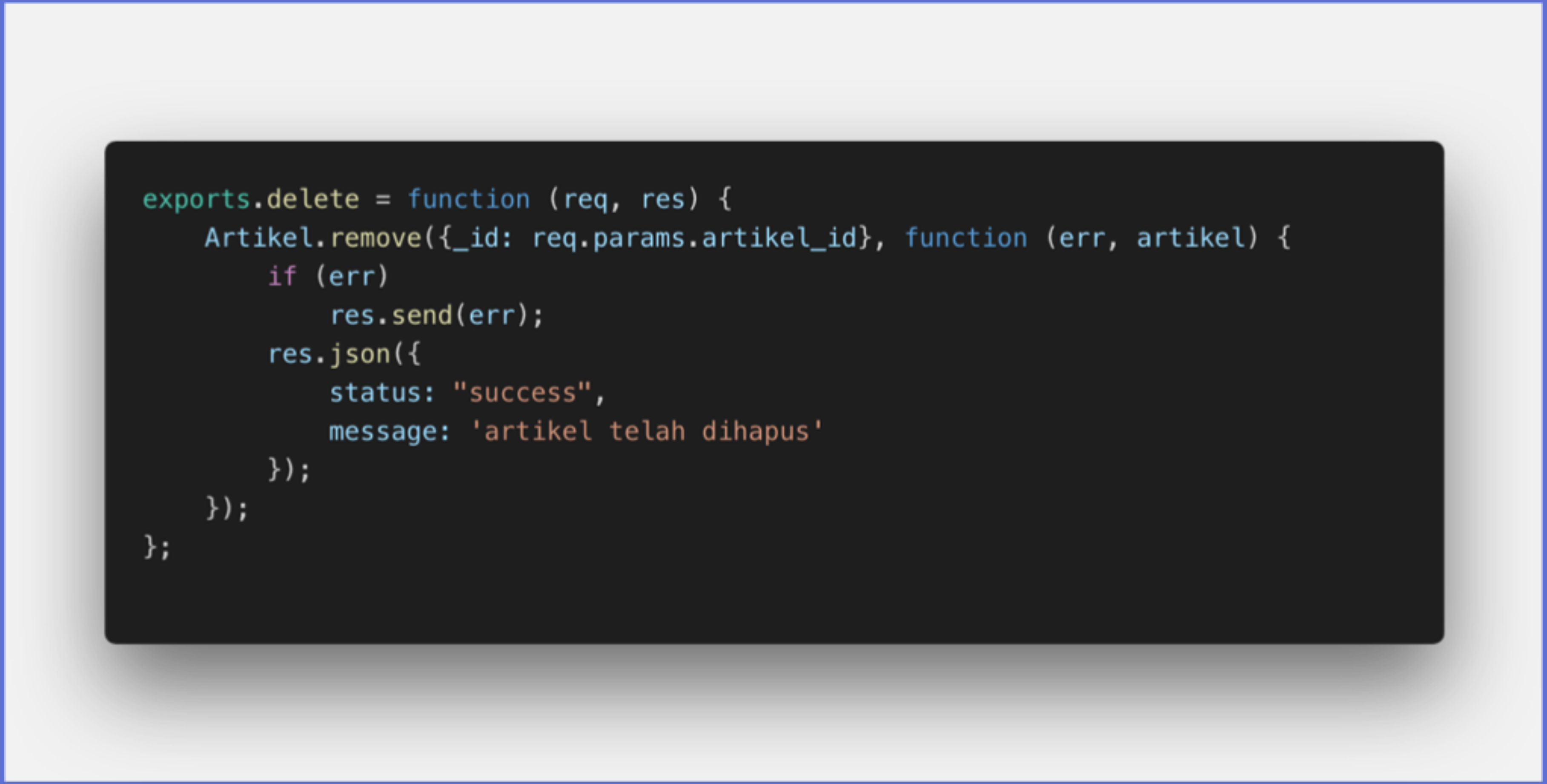
```
exports.view = function (req, res) {
    Artikel.findById(req.params.artikel_id, function (err, artikel) {
        if (err)
            res.send(err);
        res.json({
            message: 'artikel details',
            data: artikel
        });
    });
};
```

Artikel/artikelCon.js



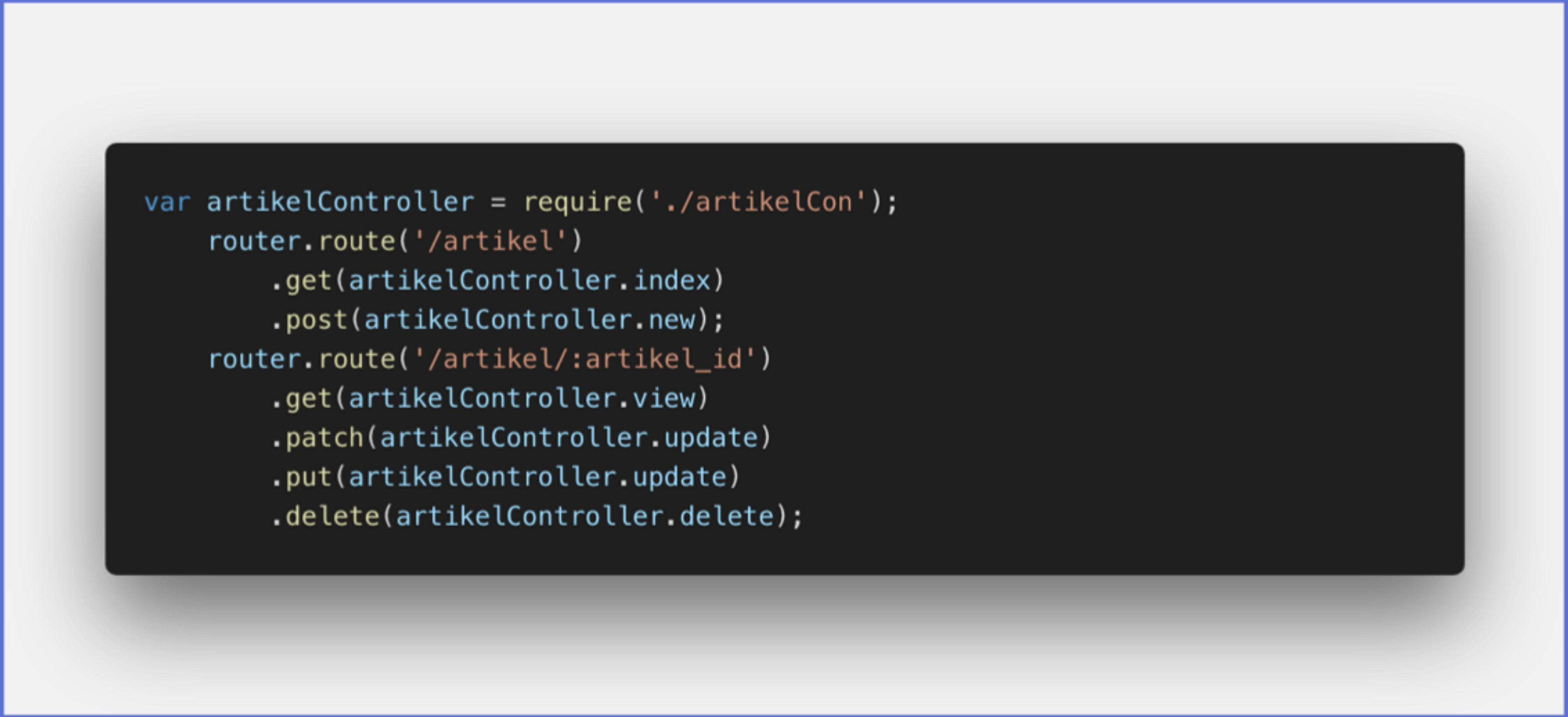
```
exports.update = function (req, res) {
        Artikel.findById(req.params.artikel_id, function (err, artikel) {
            if (err)
                res.send(err);
                artikel.judul = req.body.judul ? req.body.judul : artikel.judul;
                artikel.image = req.body.image;
                artikel.isi = req.body.isi;
                artikel.creator = req.body.creator;
            artikel.save(function (err) {
                    if (err)
                        res.json(err);
                    res.json({
                        message: 'artikel telah di update',
                        data: artikel
                    });
                });
        });
};
```

Artikel/artikelCon.js



```
exports.delete = function (req, res) {
    Artikel.remove({_id: req.params.artikel_id}, function (err, artikel) {
        if (err)
            res.send(err);
        res.json({
            status: "success",
            message: 'artikel telah dihapus'
        });
    });
};
```

Artikel/routes.js



```
var artikelController = require('./artikelCon');
    router.route('/artikel')
        .get(artikelController.index)
        .post(artikelController.new);
    router.route('/artikel/:artikel_id')
        .get(artikelController.view)
        .patch(artikelController.update)
        .put(artikelController.update)
        .delete(artikelController.delete);
```

./EOF

Sponsored by :

KEDAI DESA

KEDAI JAMAN NOW ? YA KEDAI DESA

www.kedaidesa.id | Instagram.com/kedaidesa.id