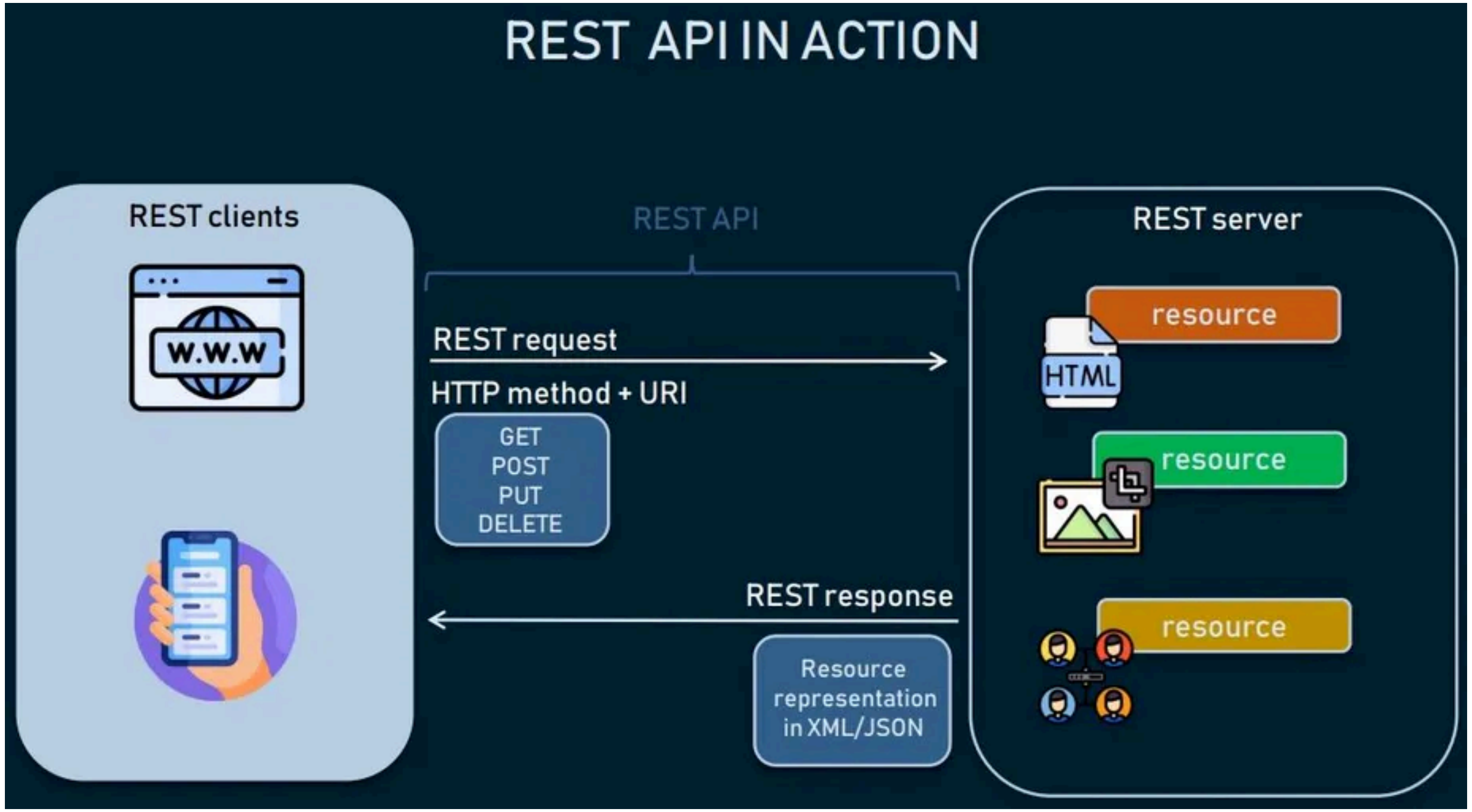
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

PHÁT TRIỂN ỨNG DỤNG WEB VỚI NODEJS (502070)

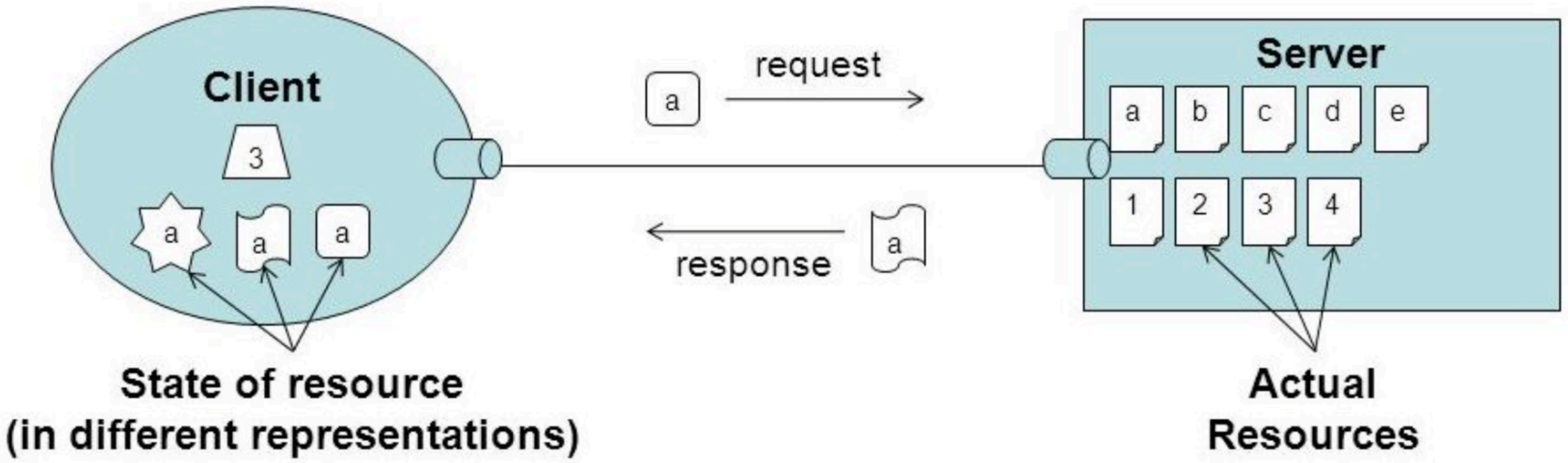
LAB5 – NODEJS

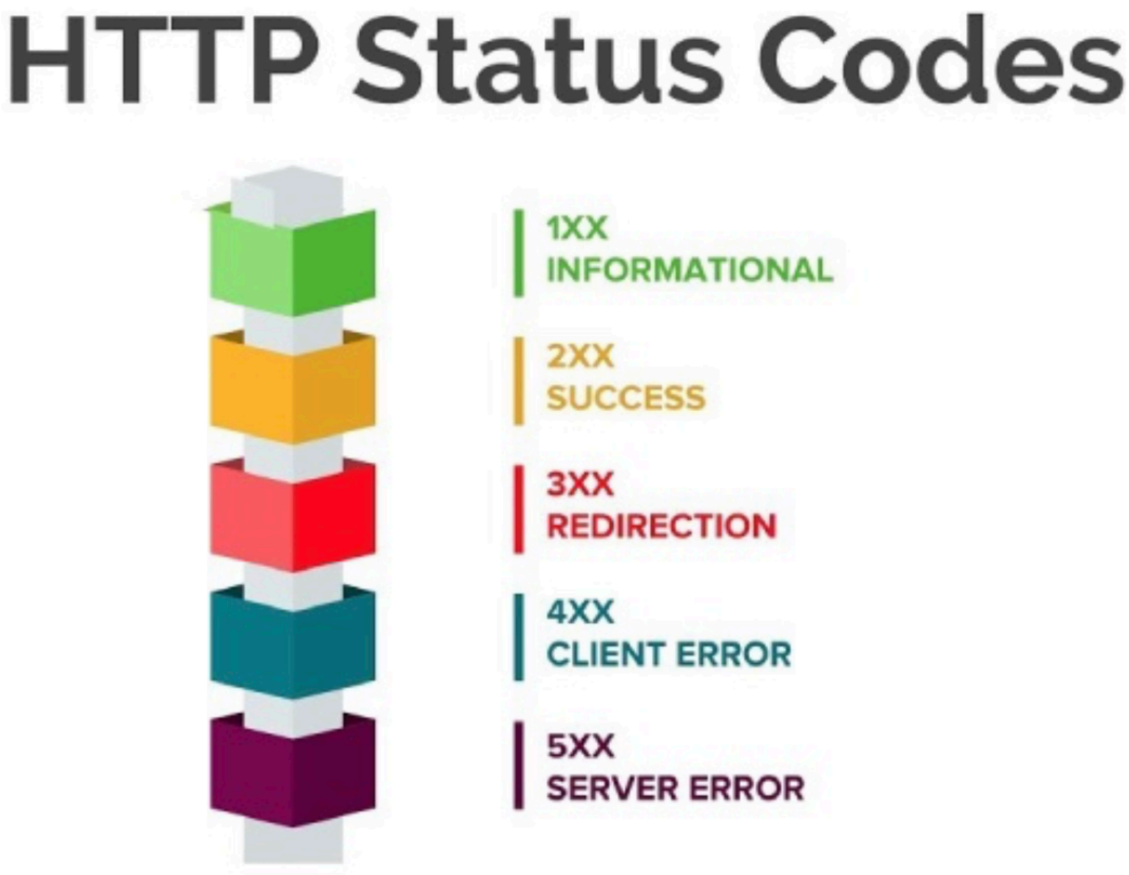
TDTU

A REST API (also known as **RESTful API**) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for **representational state transfer** and was created by computer scientist Roy Fielding.



State Transfer





HTTP Status Codes

1XX
INFORMATIONAL

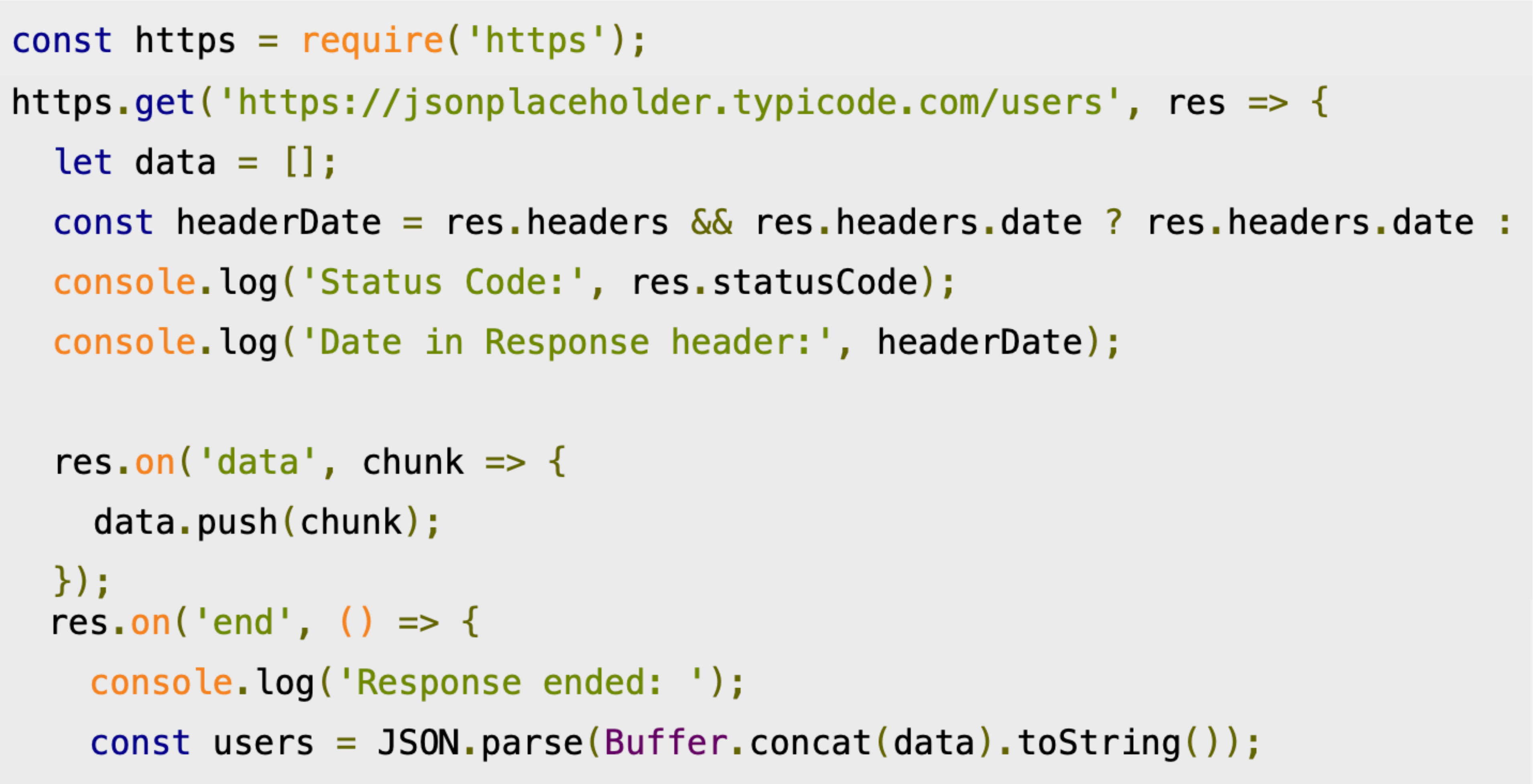
2XX
SUCCESS

3XX
REDIRECTION

4XX
CLIENT ERROR

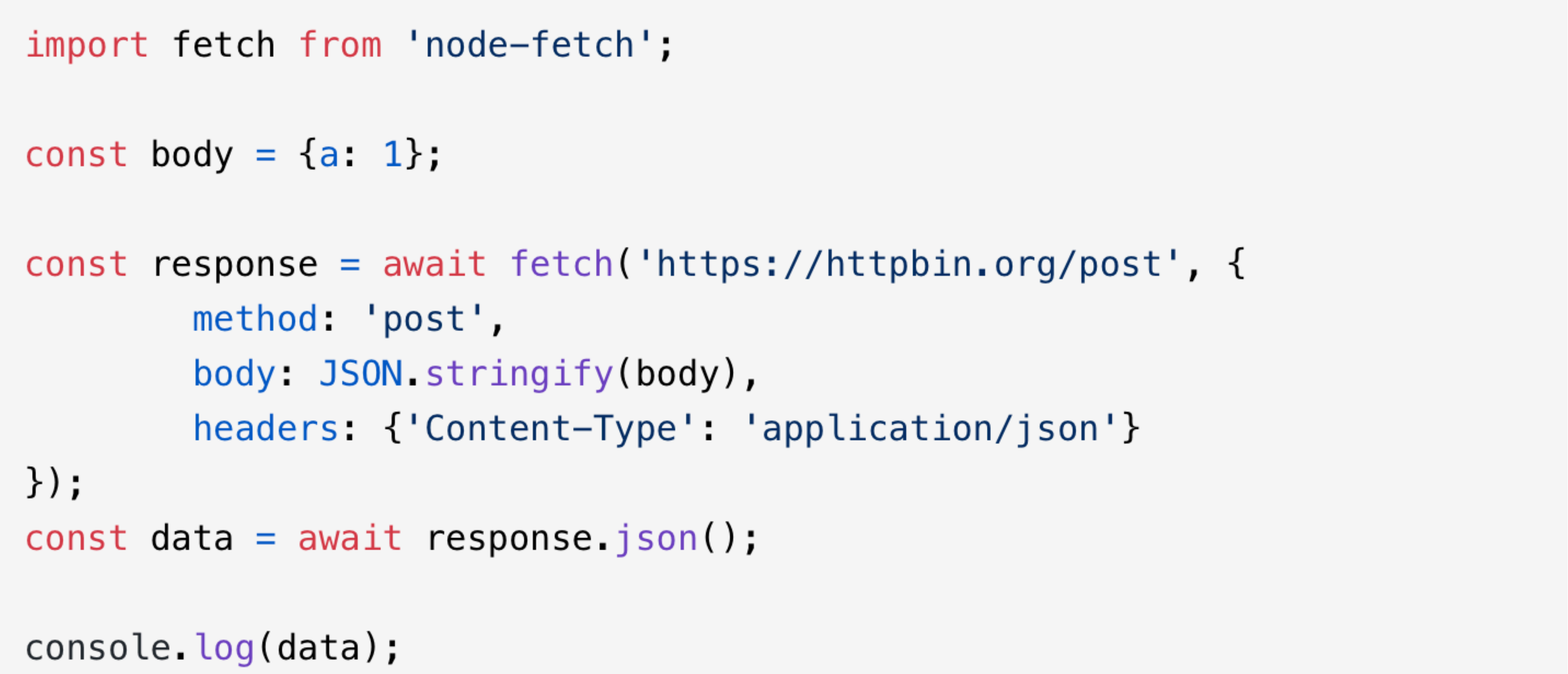
5XX
SERVER ERROR

https://nodejs.org/api/https.html



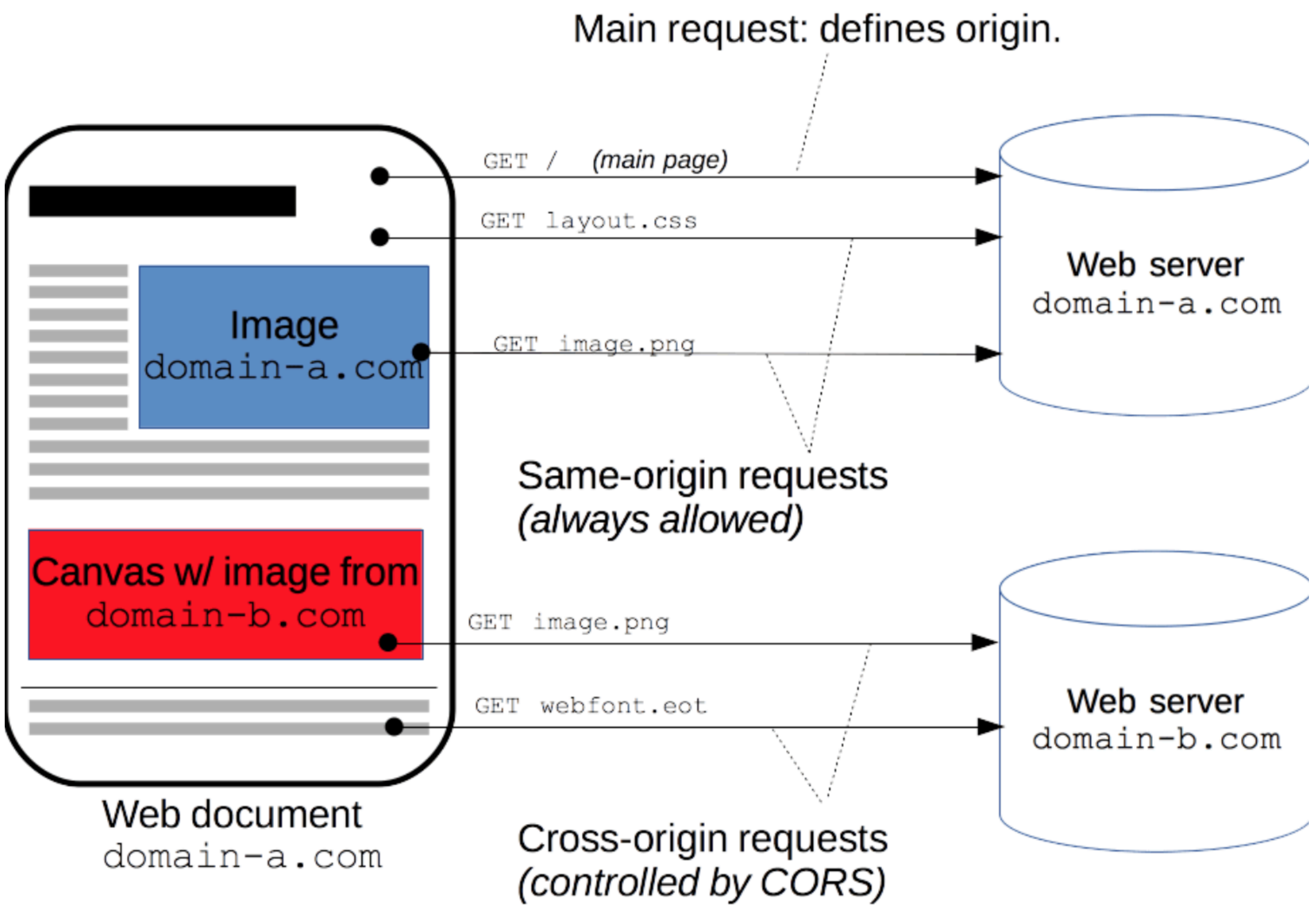
```
const https = require('https');
https.get('https://jsonplaceholder.typicode.com/users', res => {
  let data = [];
  const headerDate = res.headers && res.headers.date ? res.headers.date :
  console.log('Status Code:', res.statusCode);
  console.log('Date in Response header:', headerDate);

  res.on('data', chunk => {
    data.push(chunk);
  });
  res.on('end', () => {
    console.log('Response ended: ');
    const users = JSON.parse(Buffer.concat(data).toString());
```

https://www.npmjs.com/package/node-fetch

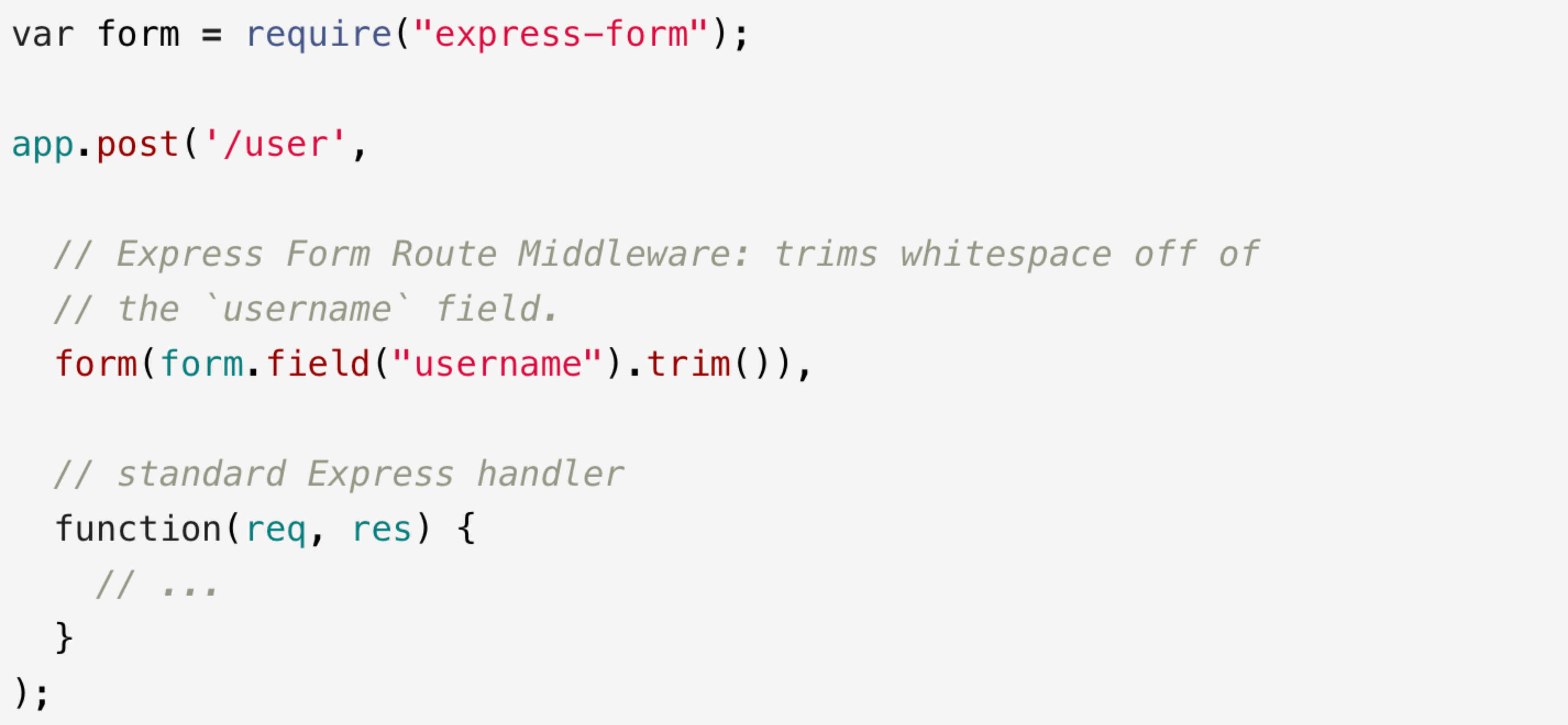


```
import fetch from 'node-fetch';

const body = {a: 1};

const response = await fetch('https://httpbin.org/post', {
        method: 'post',
        body: JSON.stringify(body),
        headers: {'Content-Type': 'application/json'}
});
const data = await response.json();

console.log(data);
```

https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

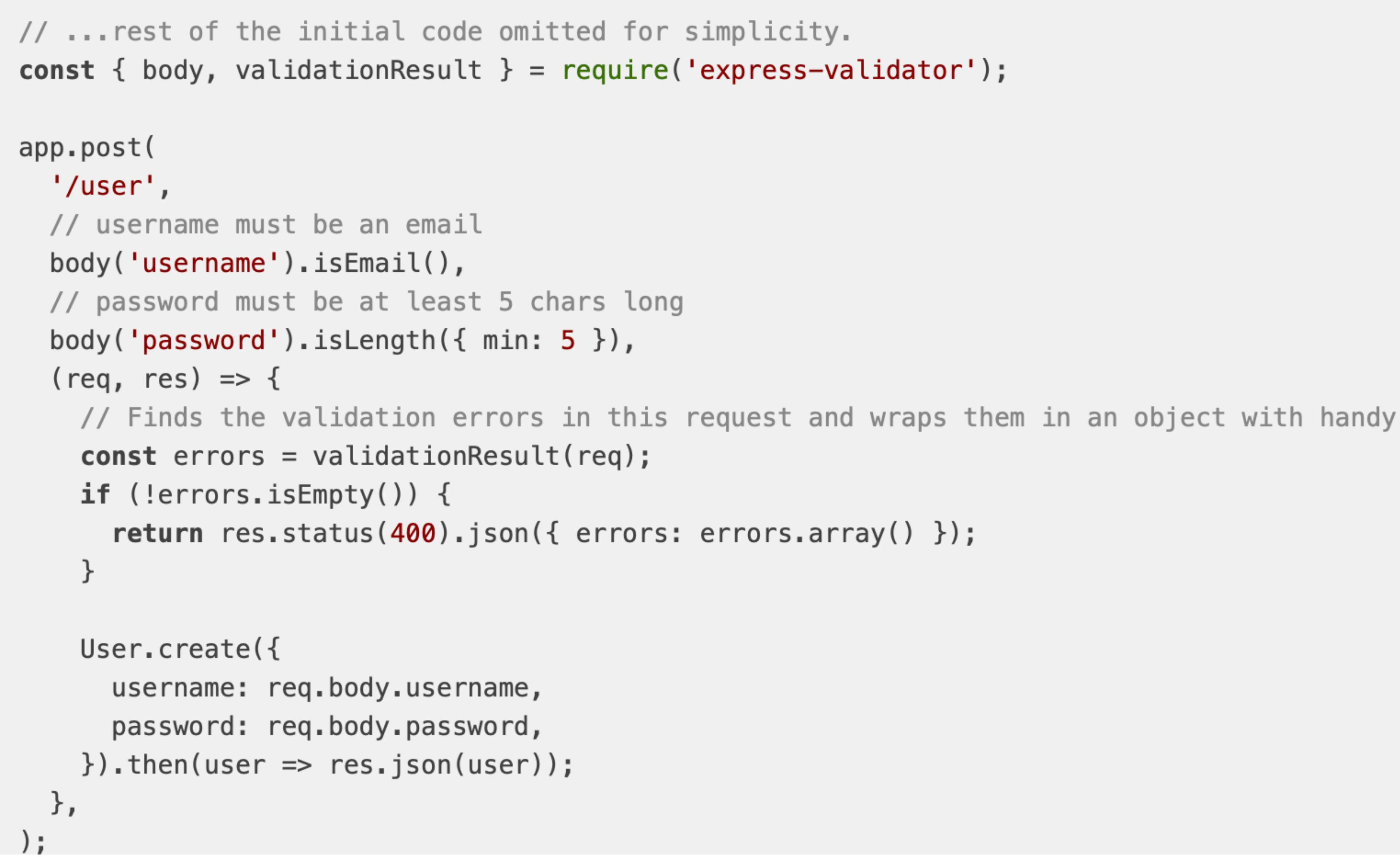


https://www.npmjs.com/package/express-form



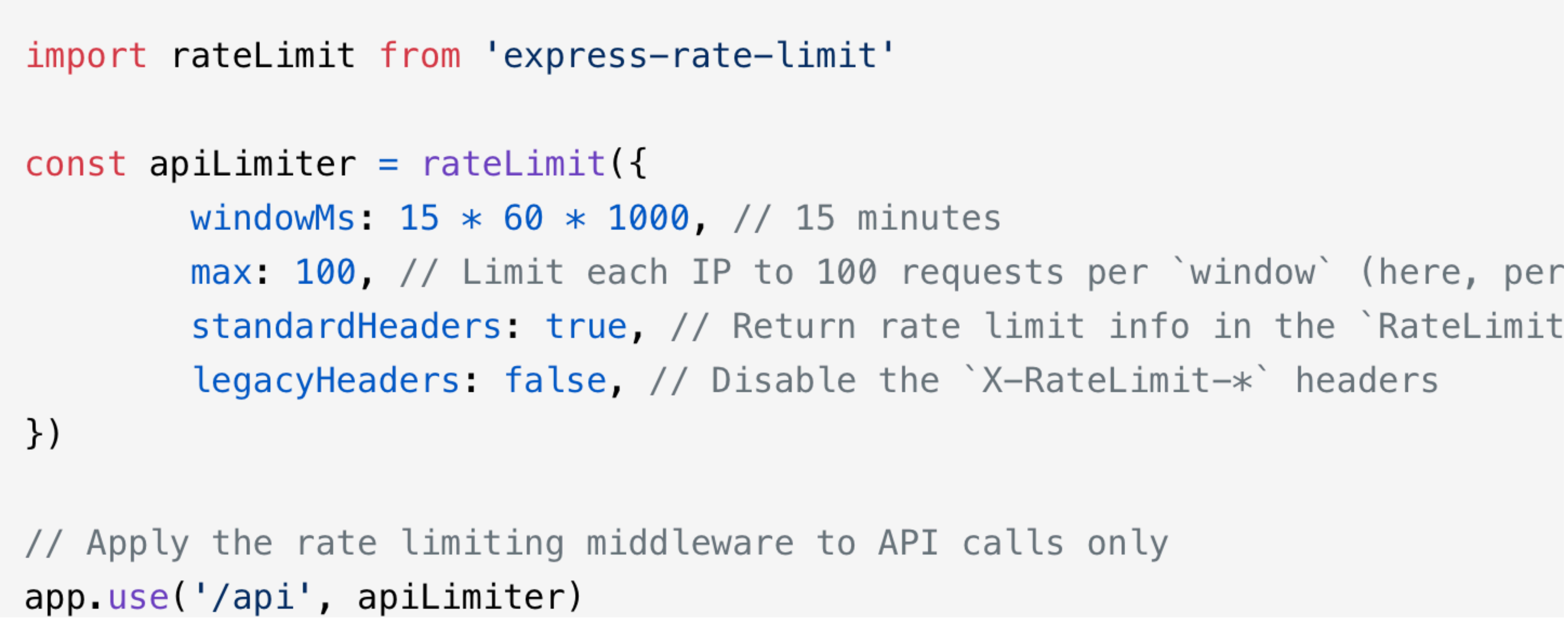
```
var form = require("express-form");

app.post('/user',

  // Express Form Route Middleware: trims whitespace off of
  // the `username` field.
  form(form.field("username").trim()),

  // standard Express handler
  function(req, res) {
    // ...
  }
);
```

https://express-validator.github.io/docs/



```
// ...rest of the initial code omitted for simplicity.
const { body, validationResult } = require('express-validator');

app.post(
  '/user',
  // username must be an email
  body('username').isEmail(),
  // password must be at least 5 chars long
  body('password').isLength({ min: 5 }),
  (req, res) => {
    // Finds the validation errors in this request and wraps them in an object with handy
    const errors = validationResult(req);
    if (!errors.isEmpty()) {
      return res.status(400).json({ errors: errors.array() });
    }

    User.create({
      username: req.body.username,
      password: req.body.password,
    }).then(user => res.json(user));
  },
);
```

https://www.npmjs.com/package/express-rate-limit
https://www.cloudflare.com/en-gb/learning/ddos/what-is-a-ddos-attack/



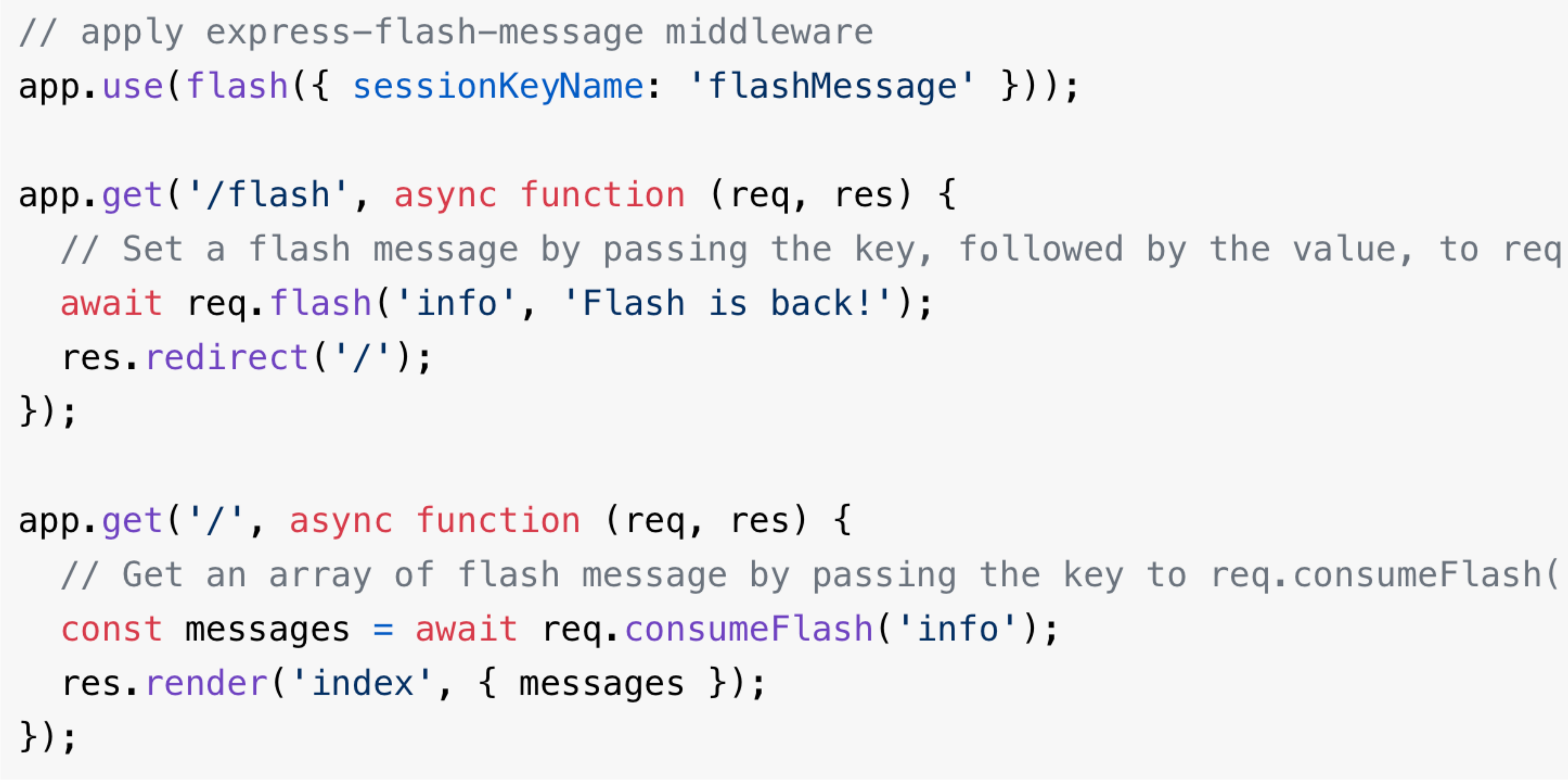
```
import rateLimit from 'express-rate-limit'

const apiLimiter = rateLimit({
        windowMs: 15 * 60 * 1000, // 15 minutes
        max: 100, // Limit each IP to 100 requests per `window` (here, per
        standardHeaders: true, // Return rate limit info in the `RateLimit
        legacyHeaders: false, // Disable the `X-RateLimit-*` headers
})

// Apply the rate limiting middleware to API calls only
app.use('/api', apiLimiter)
```

https://www.npmjs.com/package/express-flash-message

https://www.youtube.com/watch?v=OVZxrRQx46A

How to use EJS to template your Node.js application - LogRocket Blog

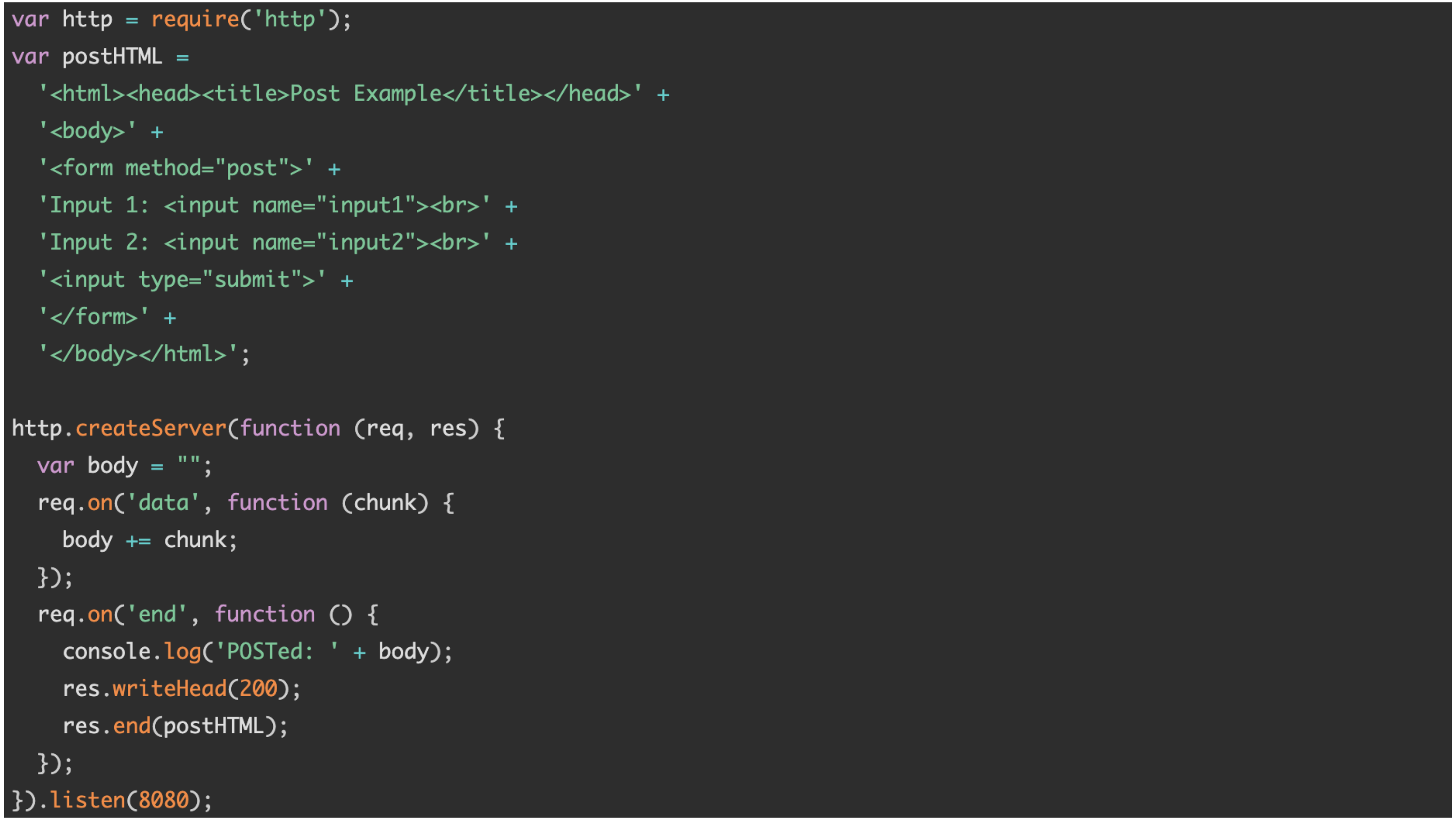
GitHub - jaredhanson/connect-flash: Flash message middleware for Connect and Express.



```
// apply express-flash-message middleware
app.use(flash({ sessionKeyName: 'flashMessage' }));

app.get('/flash', async function (req, res) {
  // Set a flash message by passing the key, followed by the value, to req
  await req.flash('info', 'Flash is back!');
  res.redirect('/');
});

app.get('/', async function (req, res) {
  // Get an array of flash message by passing the key to req.consumeFlash(
  const messages = await req.consumeFlash('info');
  res.render('index', { messages });
});
```

https://www.tutorialspoint.com/expressjs/expressjs_middleware.htm

How can I read POST data? | Node.js (nodejs.org)



```
var http = require('http');
var postHTML =
  '<html><head><title>Post Example</title></head>' +
  '<body>' +
  '<form method="post">' +
  'Input 1: <input name="input1"><br>' +
  'Input 2: <input name="input2"><br>' +
  '<input type="submit">' +
  '</form>' +
  '</body></html>';

http.createServer(function (req, res) {
  var body = "";
  req.on('data', function (chunk) {
    body += chunk;
  });
  req.on('end', function () {
    console.log('POSTed: ' + body);
    res.writeHead(200);
    res.end(postHTML);
  });
}).listen(8080);
```

Thank you