Sử dụng Express-Validator

npm install --save express-validator

Sử dụng

const { body, validationResult } = require('express-validator');

app.post('/user', [

  body('username').isEmail(),

  body('password').isLength({ min: 5 })

], (req, res) => {

  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));

});

Khi mà có lỗi thì ta sẽ nhận được object Errors như này :

{

"errors": [{

"location": "body",

"msg": "Invalid value",

"param": "username"

}]

}

Validation middlewares

Những method check validation :

* check([field, message])
* body([fields, message])
* cookie([fields, message])
* header([fields, message])
* param([fields, message])
* query([fields, message])
* checkSchema(schema)

## One Of

chỉ cần 1 điều kiện đúng là đủ

* oneOf(validationChains[, message])

const { check, oneOf, validationResult } = require('express-validator');

app.post('/start-freelancing', oneOf([

check('programming\_language').isIn(['javascript', 'java', 'php']),

check('design\_tools').isIn(['canva', 'photoshop', 'gimp'])

]), someRouteHandler);

Kết hợp OR và AND

app.post('/protected/route', oneOf([

[

check('username').exists(),

check('password').exists()

],

check('access\_token').exists()

]), someRouteHandler);

Check Combination

Kết hợp check nhiều nơi

* buildCheckFunction(locations)

**const** { buildCheckFunction } = require('express-validator');

**const** checkBodyAndQuery = buildCheckFunction(['body', 'query']);

app.put('/update-product', [

// id must be either in req.body or req.query, and must be an UUID

checkBodyAndQuery('id').isUUID()

], productUpdateHandler)

args : body, cookies, headers, params or query

Check vs body,cookie v.v

Check là bao gồm tất cả : body, cookies, headers, params, query

Còn lại nếu chỉ muốn check 1 phần thì chỉ import phần đó