Trần Xuân Thắng

**Lab 11. Viết mã lệnh tạo restful cho phương thức get để chuyển đổi ngoại tệ.**

const express = require('express');

const app = express();

app.get('/:currency/:amount', (*req*, *res*) => {

    const currency = *req*.params.currency;

    const amount = *req*.params.amount;

    const rate = {

        'usd': 22807,

        'eur': 25879,

        'jpy': 198,

    };

    const result = amount \* rate[currency];

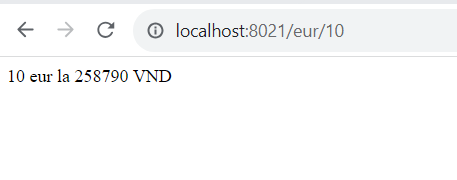
*res*.send(`${amount} ${currency} la ${result} VND`);

});

app.listen(8021);

A screenshot of a computer

Description automatically generated



A screenshot of a computer

Description automatically generated

**Lab. 12 Viết mã lệnh tao restfull cho phương thức get/post/put/delete để quản lý danh sách user.**

const express = require('express');

const app = express();

// Viết mã lệnh tao restfull cho phương thức get/post/put/delete để quản lý danh sách user

app.use(express.json());

const users = [{

        id: 1,

        name: 'Nguyen Van A',

        age: 20

    },

    {

        id: 2,

        name: 'Nguyen Van B',

        age: 21

    },

    {

        id: 3,

        name: 'Nguyen Van C',

        age: 22

    }

];

app.get('/users', (*req*, *res*) => {

*res*.send(users);

});

app.get('/users/:id', (*req*, *res*) => {

    const id = *req*.params.id;

    const user = users.find(*user* => *user*.id === parseInt(id));

    if (user) {

*res*.send(user);

    } else {

*res*.status(404).send('User not found');

    }

});

app.post('/users', (*req*, *res*) => {

    const user = {

        id: users.length + 1,

        name: *req*.body.name,

        age: *req*.body.age

    };

    users.push(user);

    const result = {

        success: true,

        message: 'Them thanh cong',

        data: users

    }

*res*.send(result);

});

app.put('/users/:id', (*req*, *res*) => {

    const id = *req*.params.id;

    const user = {

        id: parseInt(id),

        name: *req*.body.name,

        age: *req*.body.age

    }

    const index = users.findIndex(*user* => *user*.id === parseInt(id));

    if (index !== -1) {

        users[index] = user;

*res*.send({

            success: true,

            message: 'Cap nhat thanh cong',

            data: users

        });

    } else {

*res*.status(404).send({

            success: false,

            message: 'User not found'

        });

    }

});

app.delete('/users/:id', (*req*, *res*) => {

    const id = *req*.params.id;

    const index = users.findIndex(*user* => *user*.id === parseInt(id));

    if (index !== -1) {

        users.splice(index, 1);

*res*.send({

            success: true,

            message: 'Xoa thanh cong',

            data: users

        });

    } else {

*res*.status(404).send({

            success: false,

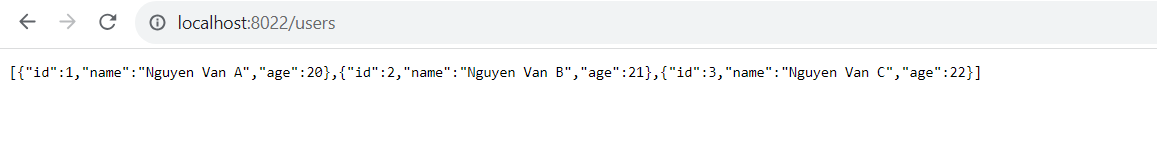
            message: 'User not found'

        });

    }

});

app.listen(8022);



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Lab 13. Viết mã lệnh để tạo web server cho các yêu cầu (req) tại cổng 8080 và trả về đáp ứng (res) là trang form web cho phép người dùng tải file lên server**

**Code**

const express = require('express');

const *fs* = require('fs');

const multer = require('multer');

const app = express();

// Viết mã lệnh để tạo web server cho các yêu cầu (req) tại cổng 8080 và trả về đáp ứng (res) là

// trang form web cho phép người dùng tải file lên server.

app.get('/', (*req*, *res*) => {

*res*.contentType('text/html');

*res*.sendFile(\_\_dirname + '/form.html');

});

const storage = multer.diskStorage({

    destination: (*req*, *file*, *cb*) => {

        cb(null, './uploads');

    },

    filename: (*req*, *file*, *cb*) => {

        cb(null, *file*.fieldname + '-' + *Date*.now());

    }

});

var upload = multer({ storage: storage });

app.post('/upload', upload.single('file'), (*req*, *res*) => {

    const file = *req*.file;

    if (!file) {

        const error = **new** *Error*('Please upload a file');

        error.httpStatusCode = 400;

        return next(error);

    }

*res*.send(file);

});

app.listen(8024);

**code form**

<!DOCTYPE *html*>

<html *lang*="en">

<head>

    <meta *charset*="UTF-8">

    <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge">

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <form *action*="upload" *method*="post" *enctype*="multipart/form-data">

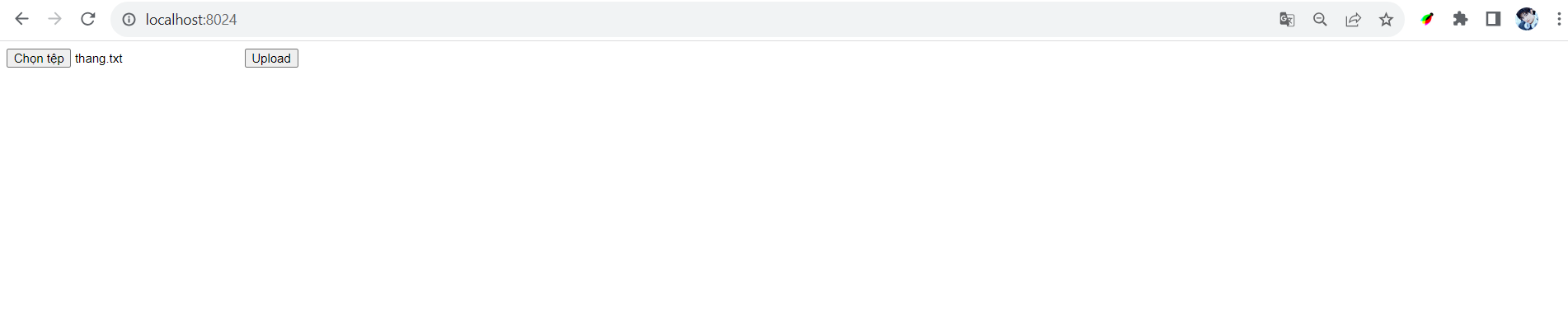
        <input *type*="file" *name*="file" *id*="file">

        <input *type*="submit" *value*="Upload">

    </form>

</body>

</html>

****

**Sau khi upload**

****