========= error.custom.js ==========

class MODEL\_NOT\_FOUND\_ERROR {

    constructor(modelName, modelId) {

        this.message = `Can't not found ${modelName} with id : ${modelId}`;

    }

}

/\* Handler \*/

const errHandlerMiddleware = async (err, req, res, next) => {

    if (err instanceof MODEL\_NOT\_FOUND\_ERROR) {

        console.error(err);

        res.render("404", { layout: false, error: err });

        return;

    }

    next(err);

};

module.exports = {

    MODEL\_NOT\_FOUND\_ERROR,

    errHandlerMiddleware,

};

========== app.js ===========

/\*

    Error Handling

    - Phải đặt ở cuối cùng sau tất cả các routes & middlewares khác

\*/

const { errHandlerMiddleware } = require("./error.custom");

app.use(errHandlerMiddleware);

========= service.js ==========

const findById = async function (id) {

    let course;

    try {

        course = await Course.findByPk(id);

    } catch (err) {

        console.error(err);

    }

    if (course === null) throw new MODEL\_NOT\_FOUND\_ERROR("Course", id);

    return course;

};

module.exports = {

    findById,

};

========= route.js ==========

router.get("/edit/:id", async (req, res, next) => {

    const id = req.params.id;

    try {

        const course = await **courseService**.findById(id);

        res.render("dashboard/course/course-form", {

            layout: "dashboard",

        });

    } catch (err) {

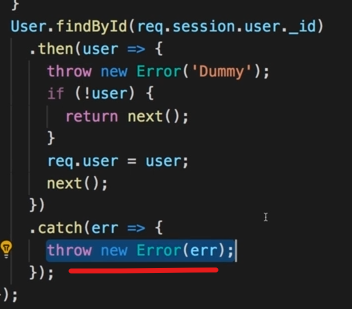
        next(err);

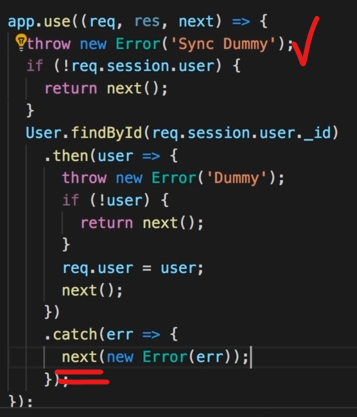
        return;

    }

});

Lưu ý khi Throw Error

Error khi throw bên trong Promise hoặc callback sẽ không trigger Express Error Middleware



Solution :

Ở bên ngoài promises, callback mình thể throw Error bình thường

Nếu bên trong Promises 🡪 phải dùng next()