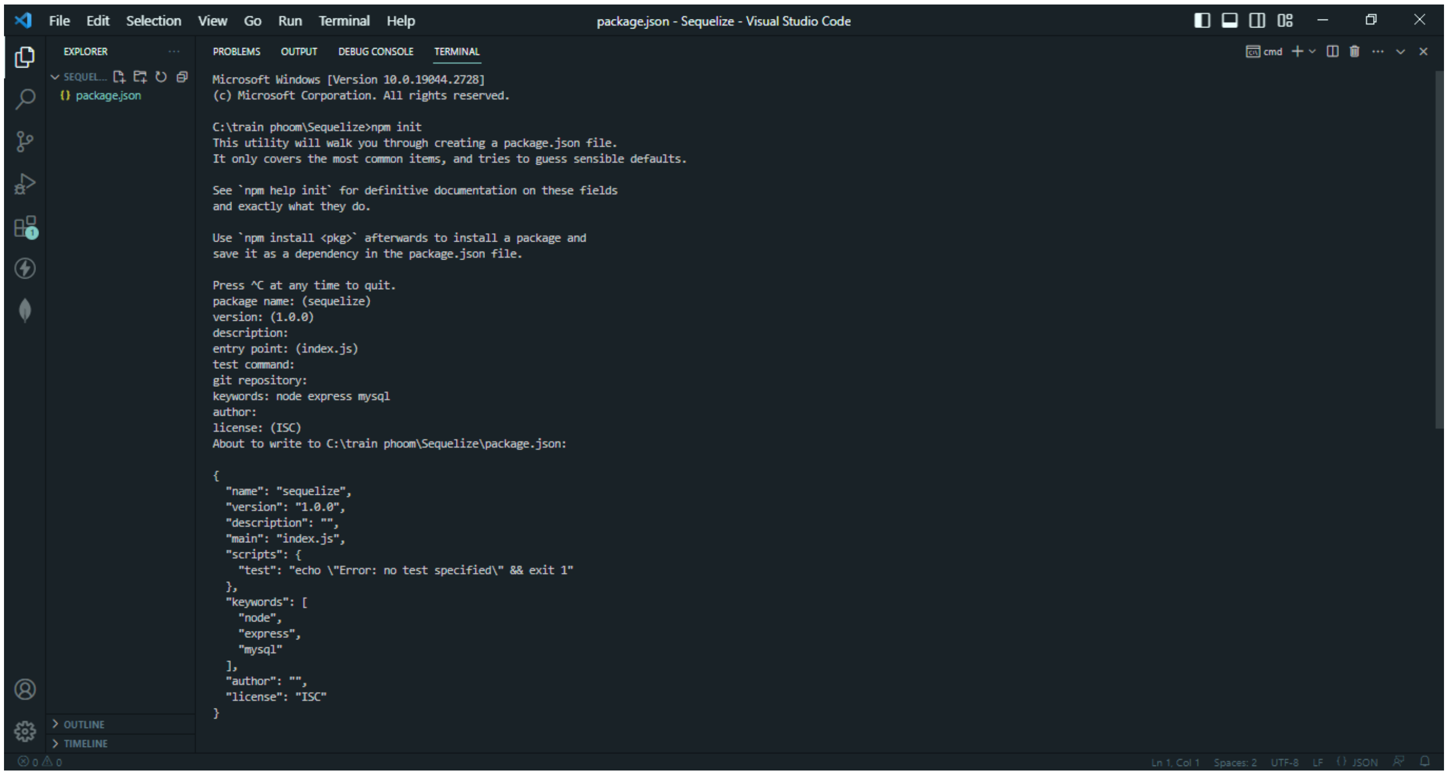
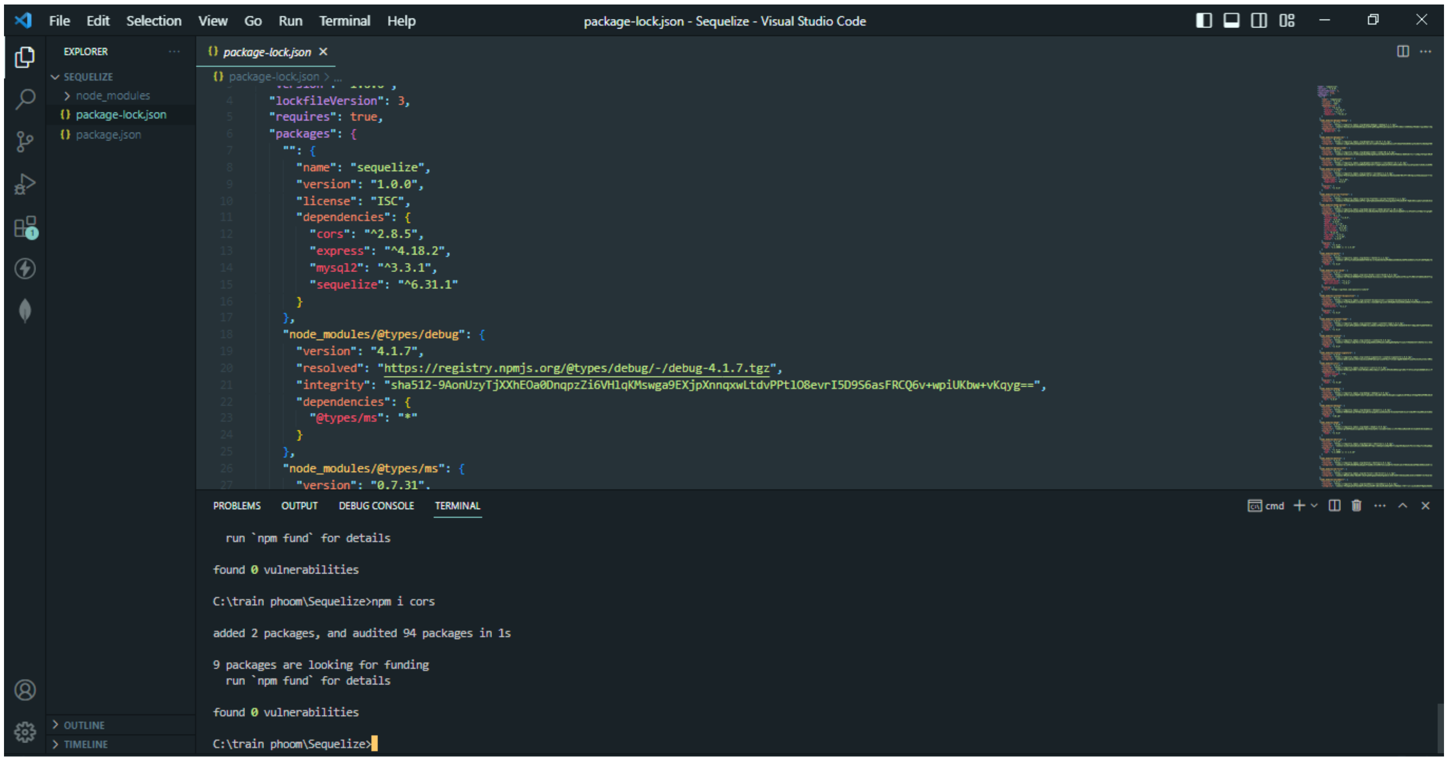
1 install node

npm init



```
Microsoft Windows [Version 10.0.19044.2728]
(c) Microsoft Corporation. All rights reserved.

C:\train phoom\Sequelize>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 init` 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: (sequelize)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords: node express mysql
author:
license: (ISC)
About to write to C:\train phoom\Sequelize\package.json:

{
  "name": "sequelize",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "node",
    "express",
    "mysql"
  ],
  "author": "",
  "license": "ISC"
}
```

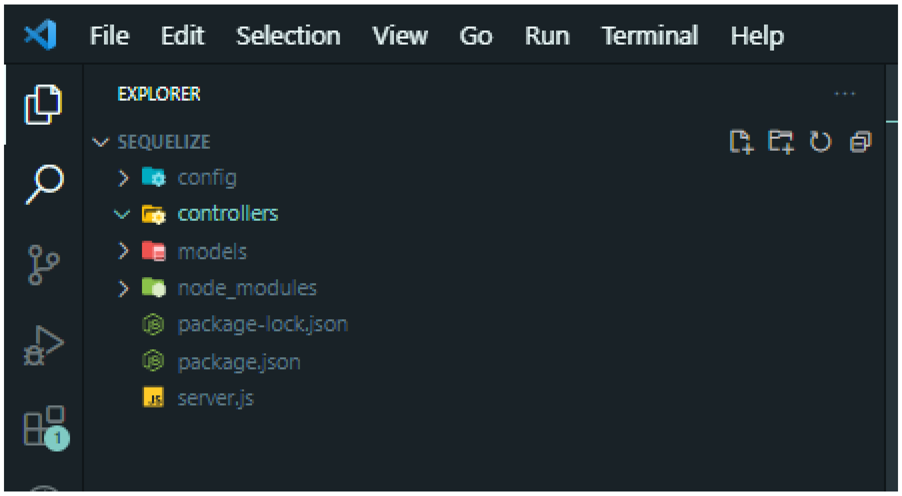
2 npm i –save express sequelize cors mysql2



```
  "lockfileVersion": 3,
  "requires": true,
  "packages": {
    "": {
      "name": "sequelize",
      "version": "1.0.0",
      "license": "ISC",
      "dependencies": {
        "cors": "^2.8.5",
        "express": "^4.18.2",
        "mysql2": "^3.3.1",
        "sequelize": "^6.31.1"
      }
    },
    "node_modules/@types/debug": {
      "version": "4.1.7",
      "resolved": "https://registry.npmjs.org/@types/debug/-/debug-4.1.7.tgz",
      "integrity": "sha512-9AonUzyTjXXhEOa0DnqpzZi6VHlqKMswga9EXjpXnnqxwLtdvPPtlO8evrI5D9S6asFRCQ6v+wpiUKbw+vKqyg==",
      "dependencies": {
        "@types/ms": "*"
      }
    },
    "node_modules/@types/ms": {
      "version": "0.7.31",
```

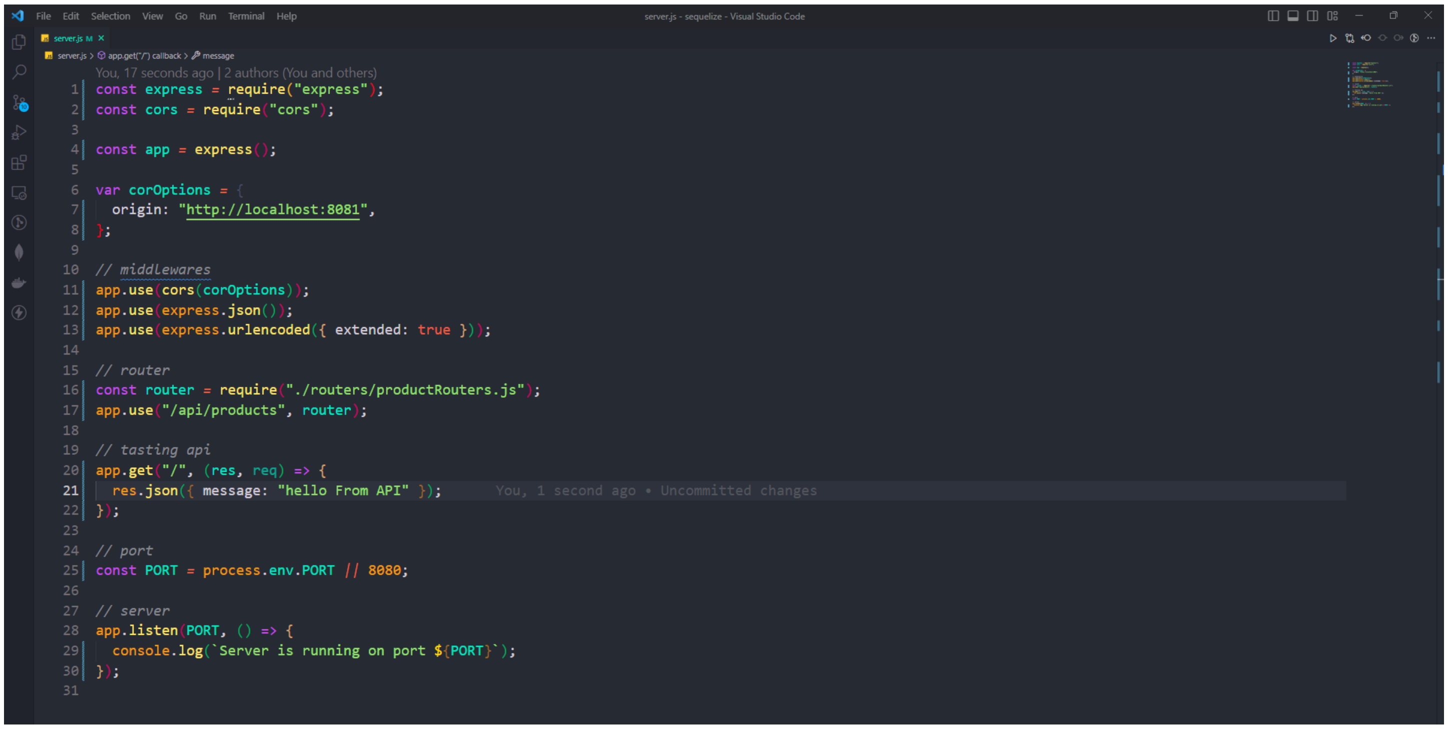
```
  run `npm fund` for details

found 0 vulnerabilities

C:\train phoom\Sequelize>npm i cors

added 2 packages, and audited 94 packages in 1s

9 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\train phoom\Sequelize>
```

3 create folder config controllers models and file server.js



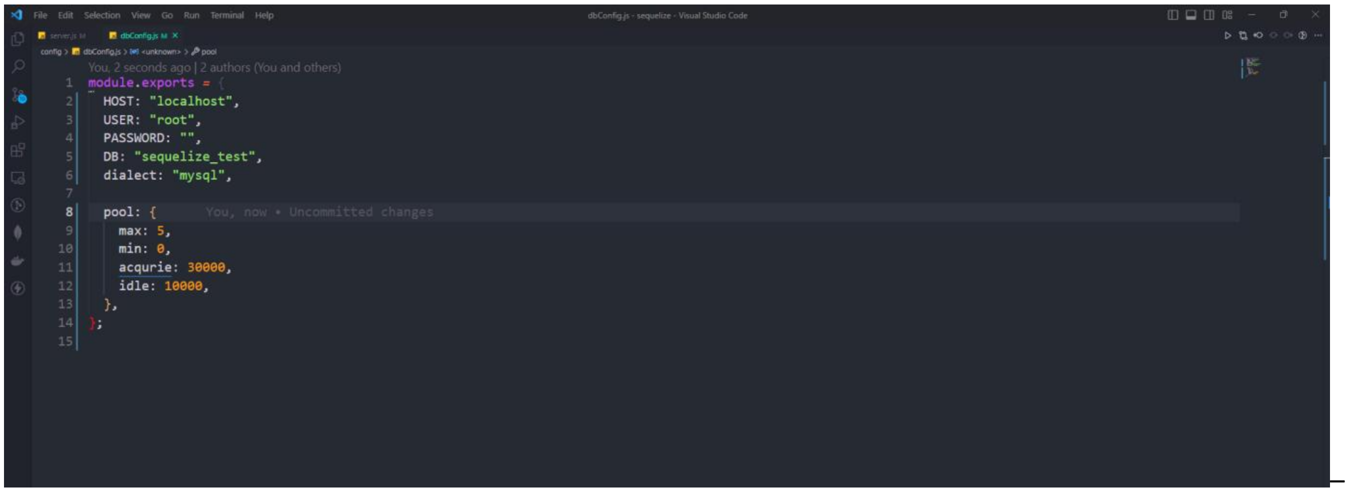
```
SEQUELIZE
  > config
  v controllers
  > models
  > node_modules
    package-lock.json
    package.json
    server.js
```

4 create file server.js


File
Edit
Selection
View
Go
Run
Terminal
Help
server.js - sequelize - Visual Studio Code
server.js M
server.js > app.get("/") callback > message
You, 17 seconds ago | 2 authors (You and others)
You, 1 second ago • Uncommitted changes

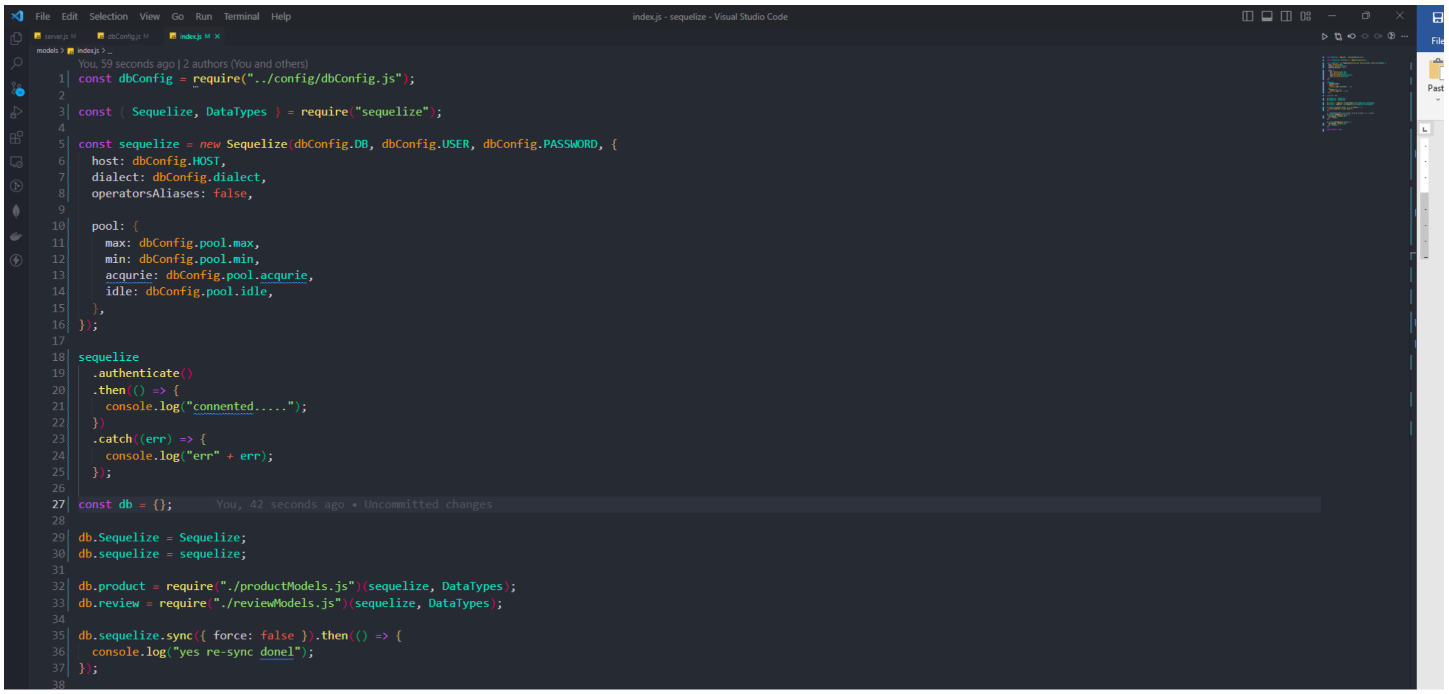
```javascript
const express = require("express");
const cors = require("cors");

const app = express();

var corOptions = {
  origin: "http://localhost:8081",
};

// middlewares
app.use(cors(corOptions));
app.use(express.json());
app.use(express.urlencoded({ extended: true }));

// router
const router = require("./routers/productRouters.js");
app.use("/api/products", router);

// tasting api
app.get("/", (res, req) => {
  res.json({ message: "hello From API" });
});

// port
const PORT = process.env.PORT || 8080;

// server
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

5 create file dbConfig.js in folder config


File
Edit
Selection
View
Go
Run
Terminal
Help
dbConfig.js - sequelize - Visual Studio Code
server.js M
dbConfig.js M
config > dbConfig.js > <unknown> > pool
You, 2 seconds ago | 2 authors (You and others)
You, now • Uncommitted changes

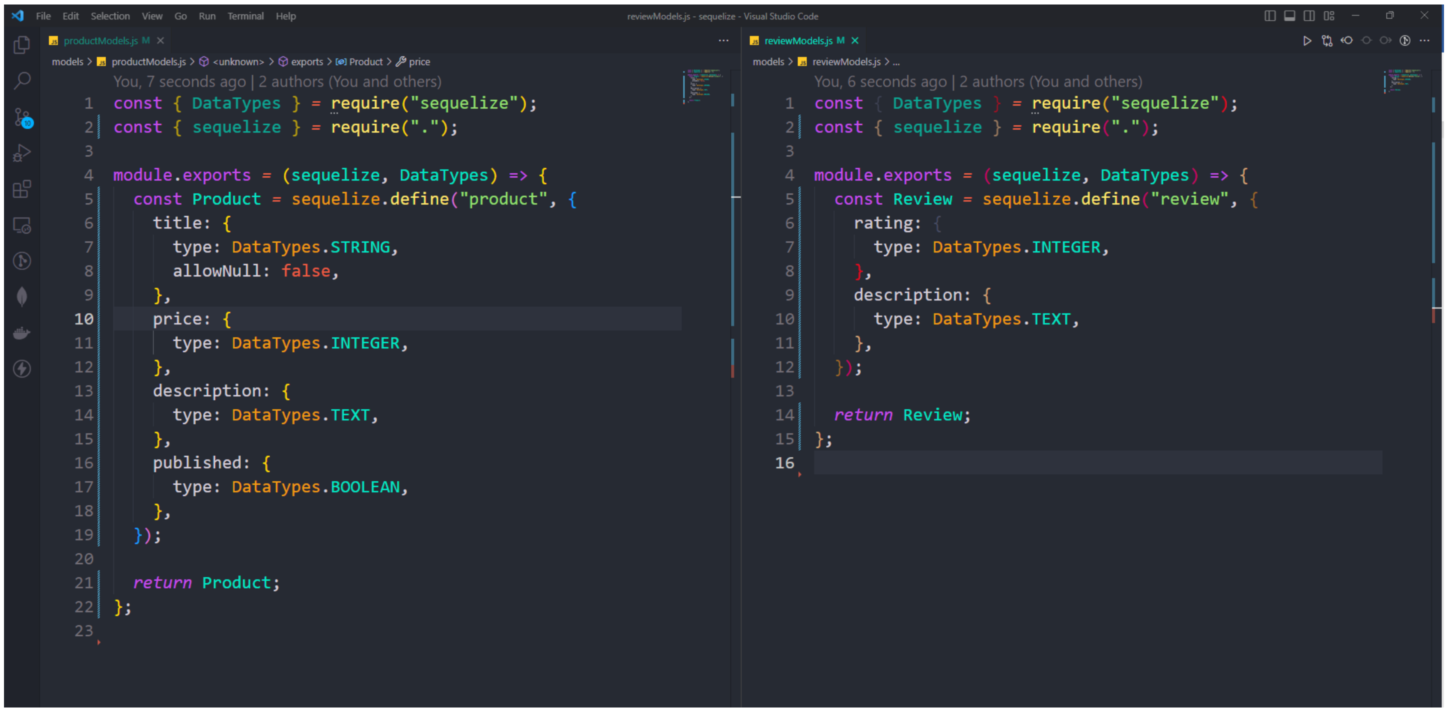
```javascript
module.exports = {
  HOST: "localhost",
  USER: "root",
  PASSWORD: "",
  DB: "sequelize_test",
  dialect: "mysql",

  pool: {
    max: 5,
    min: 0,
    acqurie: 30000,
    idle: 10000,
  },
};
```

6 create file index.js in folder models



```
const dbConfig = require("../config/dbConfig.js");

const { Sequelize, DataTypes } = require("sequelize");

const sequelize = new Sequelize(dbConfig.DB, dbConfig.USER, dbConfig.PASSWORD, {
  host: dbConfig.HOST,
  dialect: dbConfig.dialect,
  operatorsAliases: false,

  pool: {
    max: dbConfig.pool.max,
    min: dbConfig.pool.min,
    acqurie: dbConfig.pool.acqurie,
    idle: dbConfig.pool.idle,
  },
});

sequelize
  .authenticate()
  .then(() => {
    console.log("connented.....");
  })
  .catch((err) => {
    console.log("err" + err);
  });

const db = {};

db.Sequelize = Sequelize;
db.sequelize = sequelize;

db.product = require("./productModels.js")(sequelize, DataTypes);
db.review = require("./reviewModels.js")(sequelize, DataTypes);

db.sequelize.sync({ force: false }).then(() => {
  console.log("yes re-sync donel");
});
```

7 create file product and review in folder models

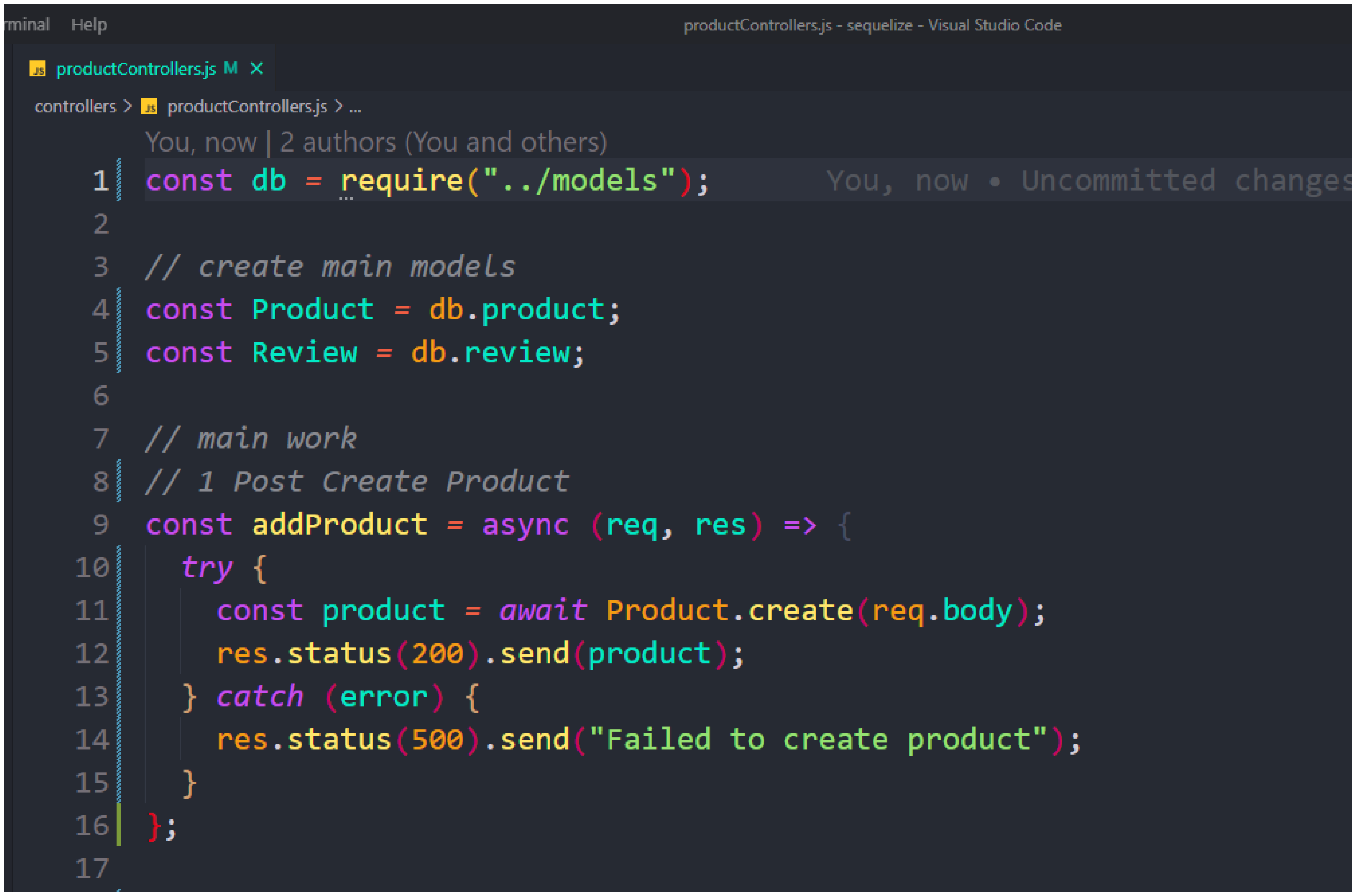


```
const { DataTypes } = require("sequelize");
const { sequelize } = require(".");

module.exports = (sequelize, DataTypes) => {
  const Product = sequelize.define("product", {
    title: {
      type: DataTypes.STRING,
      allowNull: false,
    },
    price: {
      type: DataTypes.INTEGER,
    },
    description: {
      type: DataTypes.TEXT,
    },
    published: {
      type: DataTypes.BOOLEAN,
    },
  });

  return Product;
};
```

```
const { DataTypes } = require("sequelize");
const { sequelize } = require(".");

module.exports = (sequelize, DataTypes) => {
  const Review = sequelize.define("review", {
    rating: {
      type: DataTypes.INTEGER,
    },
    description: {
      type: DataTypes.TEXT,
    },
  });

  return Review;
};
```

8 create productcontrollers.js

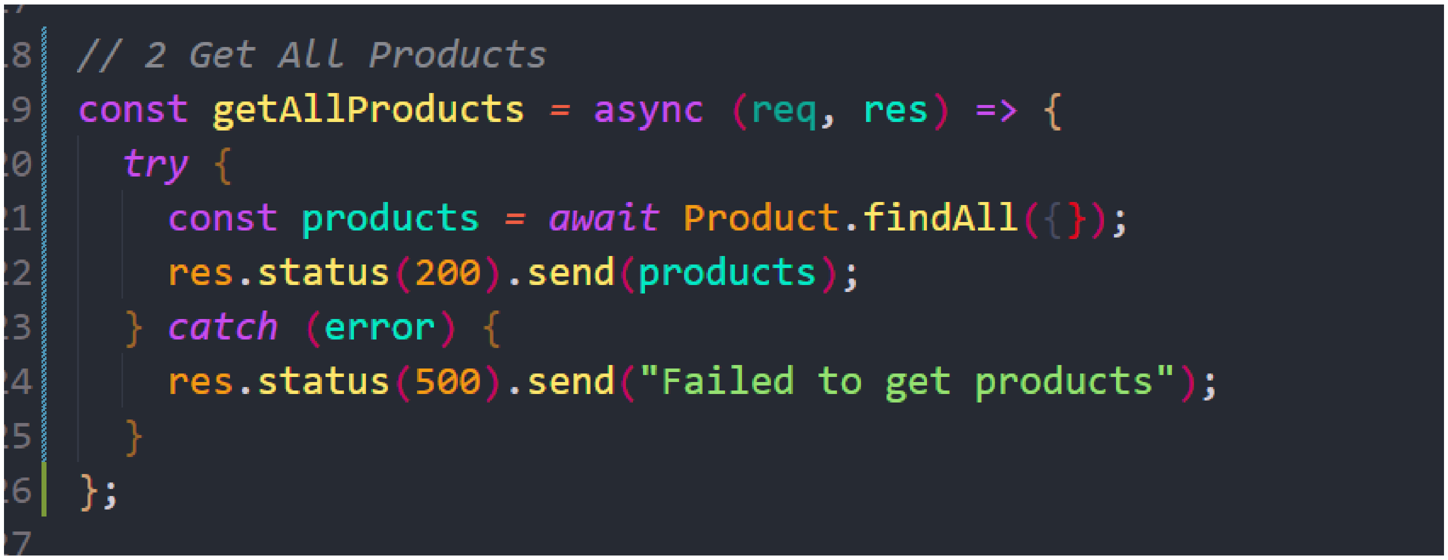
8.1 create product



rminal Help
productControllers.js - sequelize - Visual Studio Code
productControllers.js M
controllers > productControllers.js > ...
You, now | 2 authors (You and others)
You, now • Uncommitted changes

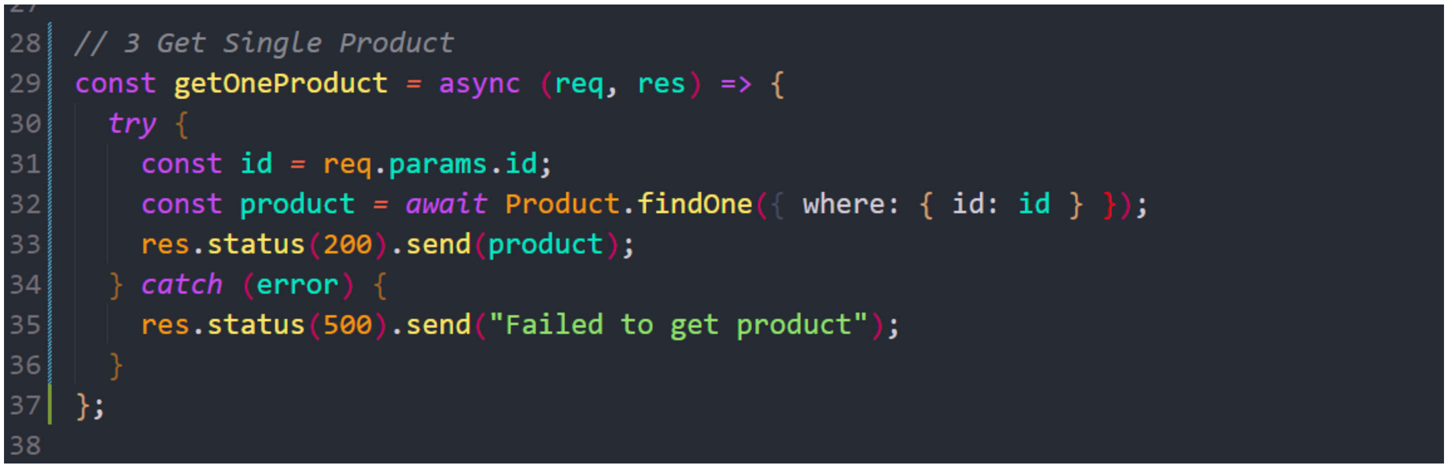
```javascript
const db = require("../models");

// create main models
const Product = db.product;
const Review = db.review;

// main work
// 1 Post Create Product
const addProduct = async (req, res) => {
  try {
    const product = await Product.create(req.body);
    res.status(200).send(product);
  } catch (error) {
    res.status(500).send("Failed to create product");
  }
};
```

8.2 get all product



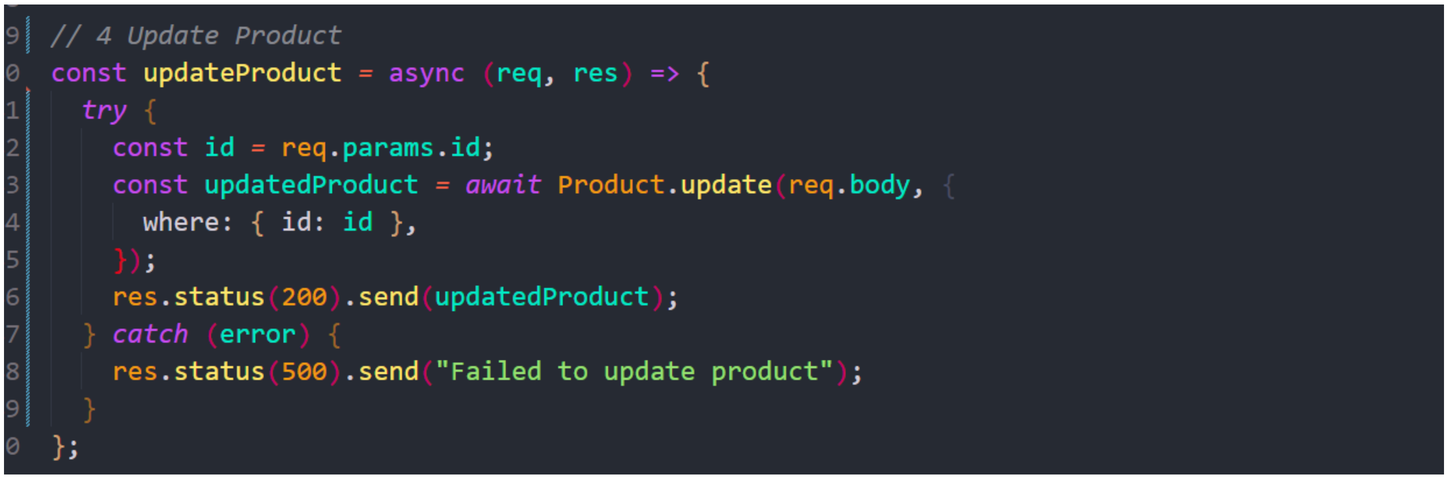
```javascript
// 2 Get All Products
const getAllProducts = async (req, res) => {
  try {
    const products = await Product.findAll({});
    res.status(200).send(products);
  } catch (error) {
    res.status(500).send("Failed to get products");
  }
};
```

8.3 get single product



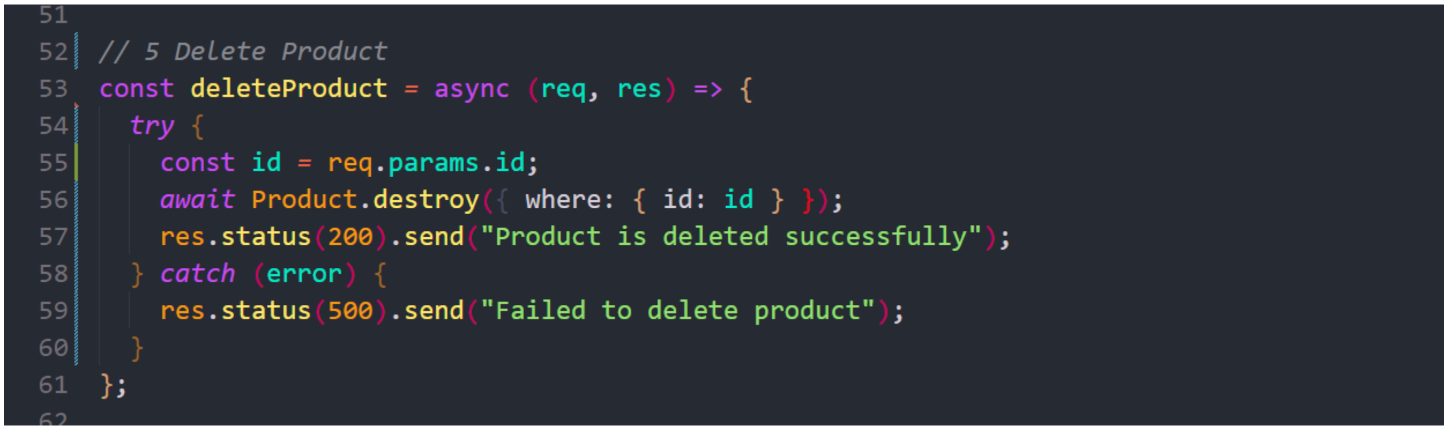
```
// 3 Get Single Product
const getOneProduct = async (req, res) => {
  try {
    const id = req.params.id;
    const product = await Product.findOne({ where: { id: id } });
    res.status(200).send(product);
  } catch (error) {
    res.status(500).send("Failed to get product");
  }
};
```

8.4 update Product



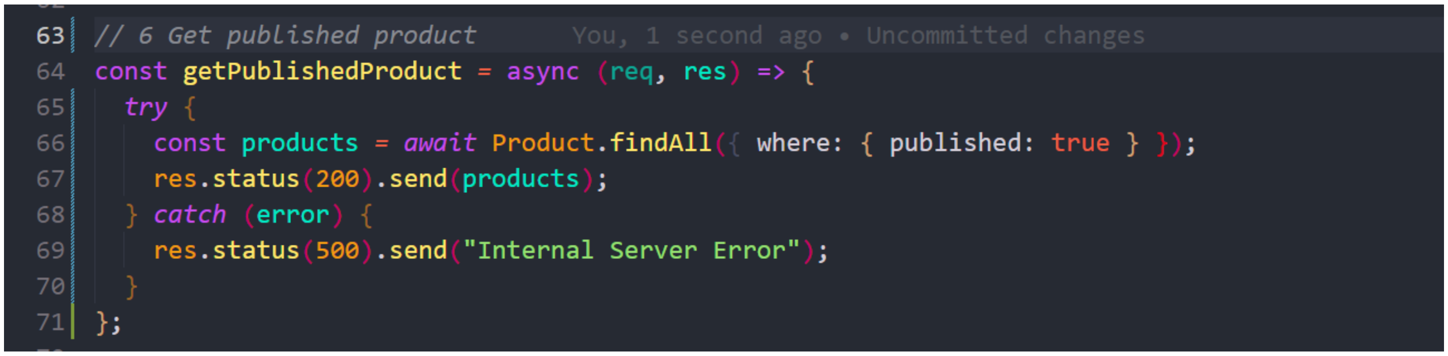
```
// 4 Update Product
const updateProduct = async (req, res) => {
  try {
    const id = req.params.id;
    const updatedProduct = await Product.update(req.body, {
      where: { id: id },
    });
    res.status(200).send(updatedProduct);
  } catch (error) {
    res.status(500).send("Failed to update product");
  }
};
```

8.5 delete product



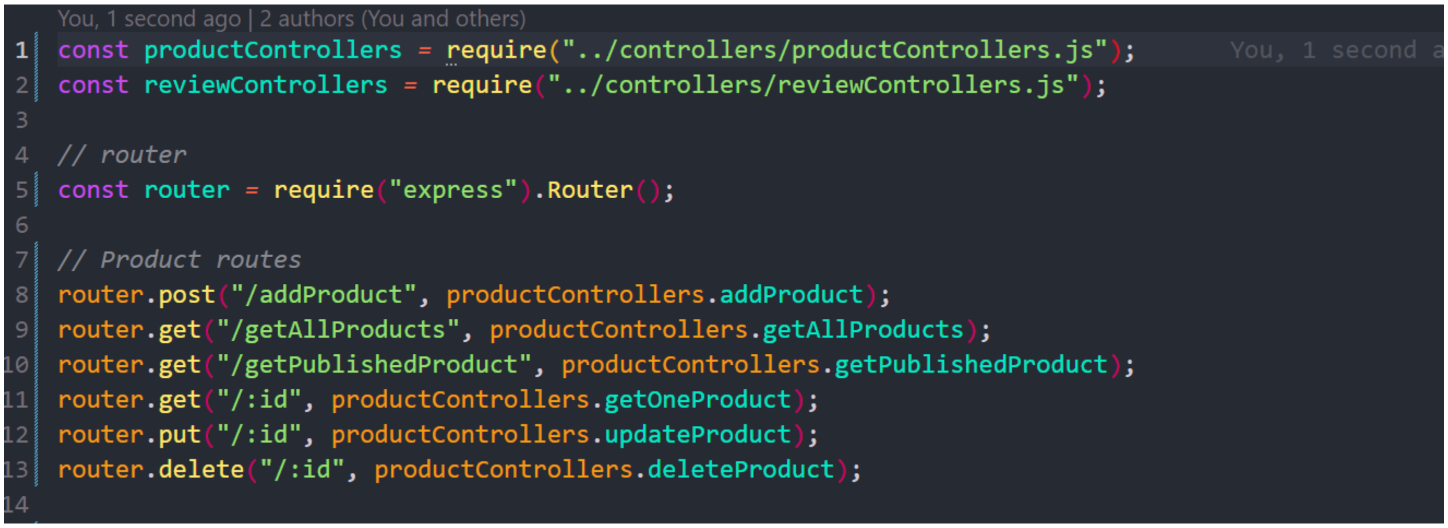
```
// 5 Delete Product
const deleteProduct = async (req, res) => {
  try {
    const id = req.params.id;
    await Product.destroy({ where: { id: id } });
    res.status(200).send("Product is deleted successfully");
  } catch (error) {
    res.status(500).send("Failed to delete product");
  }
};
```

8.6 *published product*



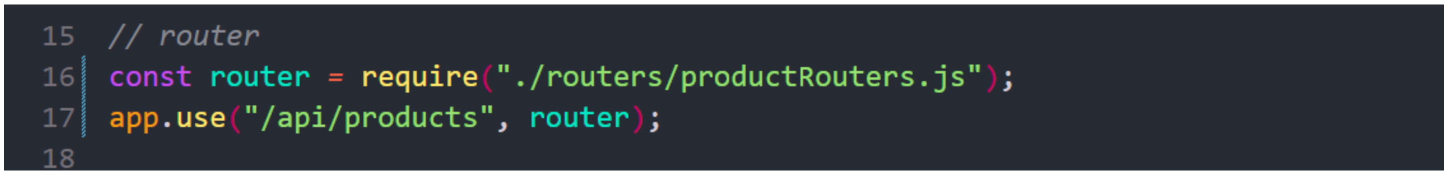
```
// 6 Get published product      You, 1 second ago • Uncommitted changes
const getPublishedProduct = async (req, res) => {
  try {
    const products = await Product.findAll({ where: { published: true } });
    res.status(200).send(products);
  } catch (error) {
    res.status(500).send("Internal Server Error");
  }
};
```

9 create folder Routers and create file productRouters and create router in file productRouters.js



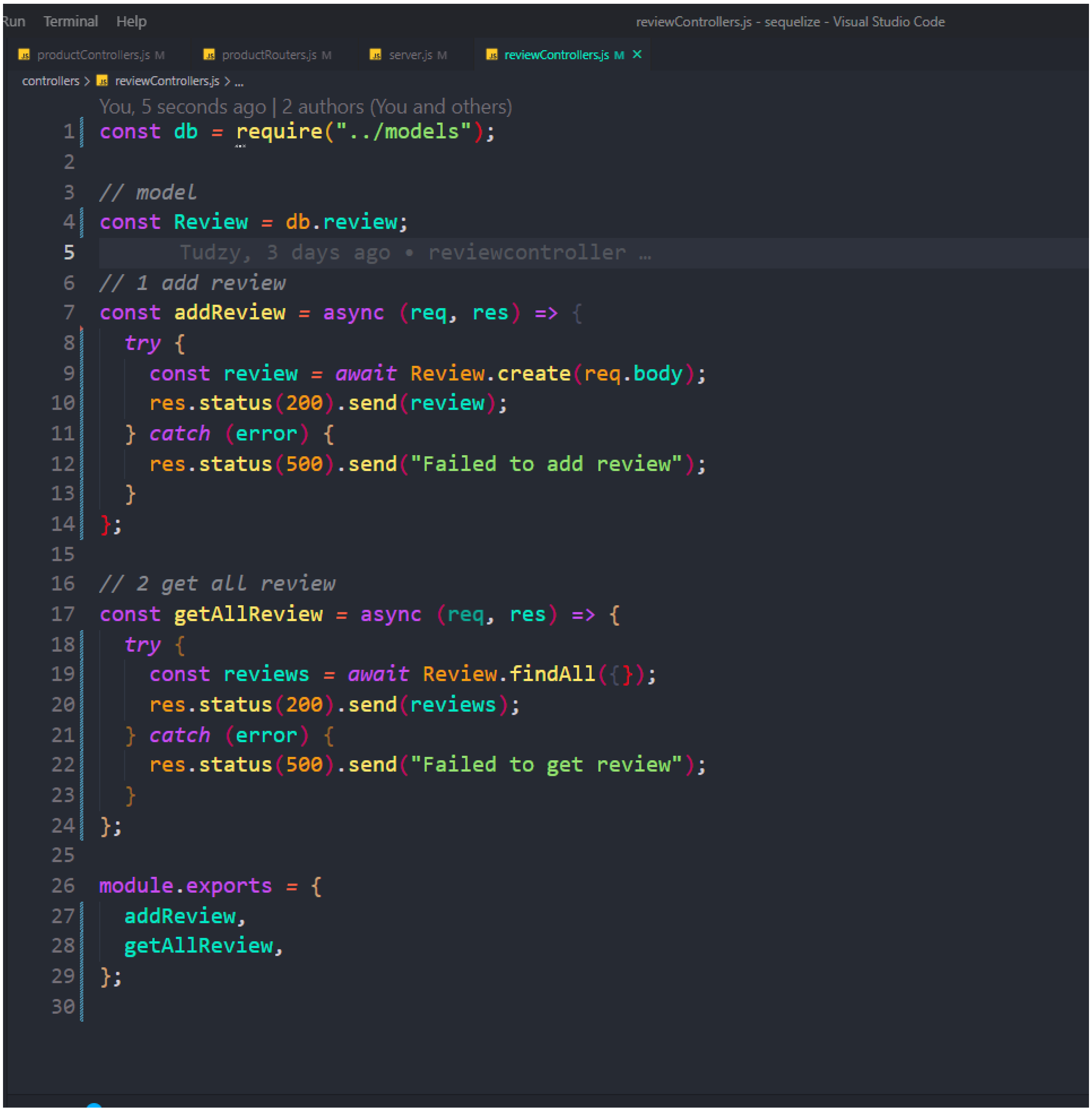
```
You, 1 second ago | 2 authors (You and others)
const productControllers = require("../controllers/productControllers.js");      You, 1 second a
const reviewControllers = require("../controllers/reviewControllers.js");

// router
const router = require("express").Router();

// Product routes
router.post("/addProduct", productControllers.addProduct);
router.get("/getAllProducts", productControllers.getAllProducts);
router.get("/getPublishedProduct", productControllers.getPublishedProduct);
router.get("/:id", productControllers.getOneProduct);
router.put("/:id", productControllers.updateProduct);
router.delete("/:id", productControllers.deleteProduct);
```

10 create router in file server.js



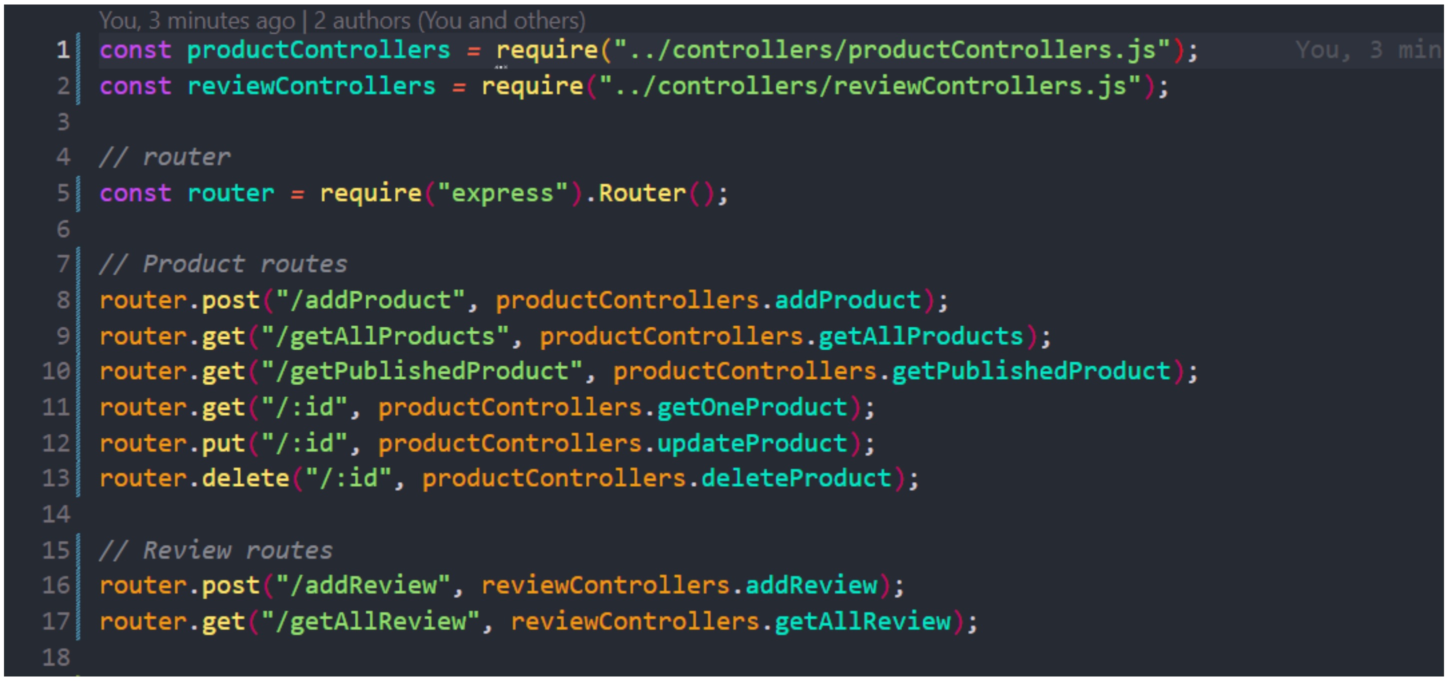
```
// router
const router = require("./routers/productRouters.js");
app.use("/api/products", router);
```

11 create file review in folder controllers


Run Terminal Help
reviewControllers.js - sequelize - Visual Studio Code
productControllers.js M
productRouters.js M
server.js M
reviewControllers.js M
controllers > reviewControllers.js > ...
You, 5 seconds ago | 2 authors (You and others)
Tudzy, 3 days ago • reviewcontroller ...

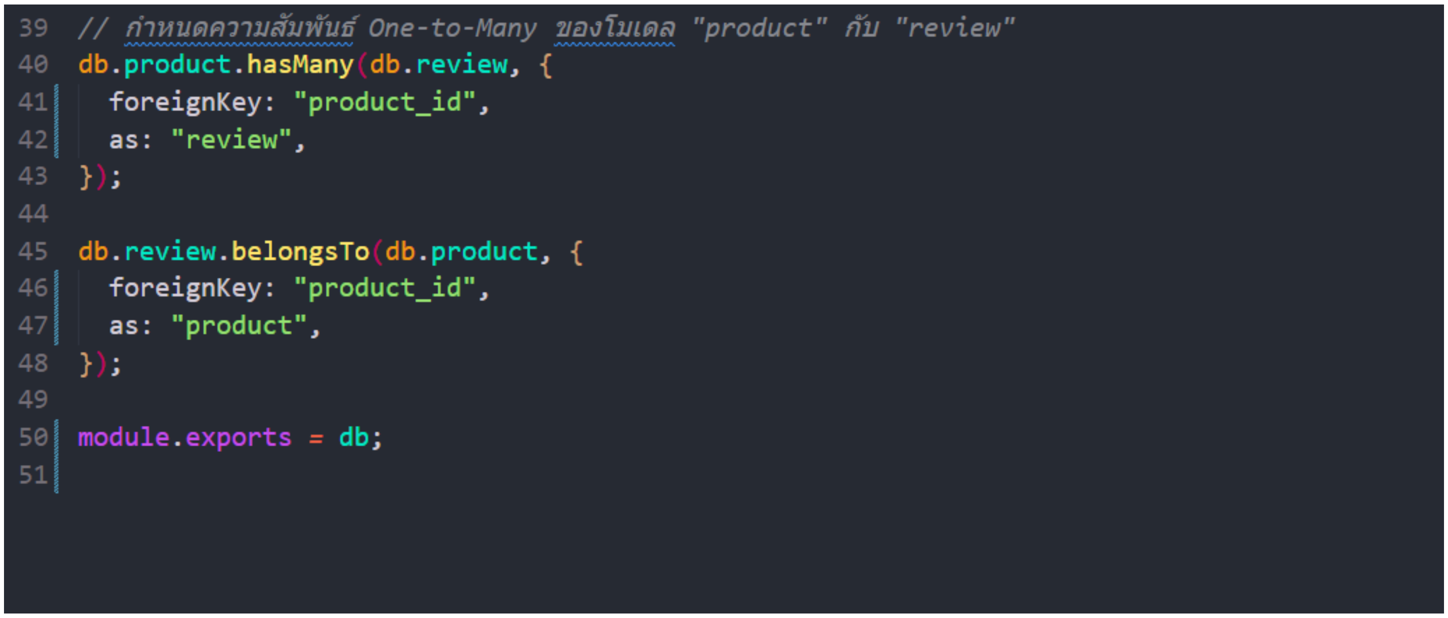
```javascript
const db = require("../models");

// model
const Review = db.review;

// 1 add review
const addReview = async (req, res) => {
  try {
    const review = await Review.create(req.body);
    res.status(200).send(review);
  } catch (error) {
    res.status(500).send("Failed to add review");
  }
};

// 2 get all review
const getAllReview = async (req, res) => {
  try {
    const reviews = await Review.findAll({});
    res.status(200).send(reviews);
  } catch (error) {
    res.status(500).send("Failed to get review");
  }
};

module.exports = {
  addReview,
  getAllReview,
};
```

12 add reviewcontrollers.js in productRouter



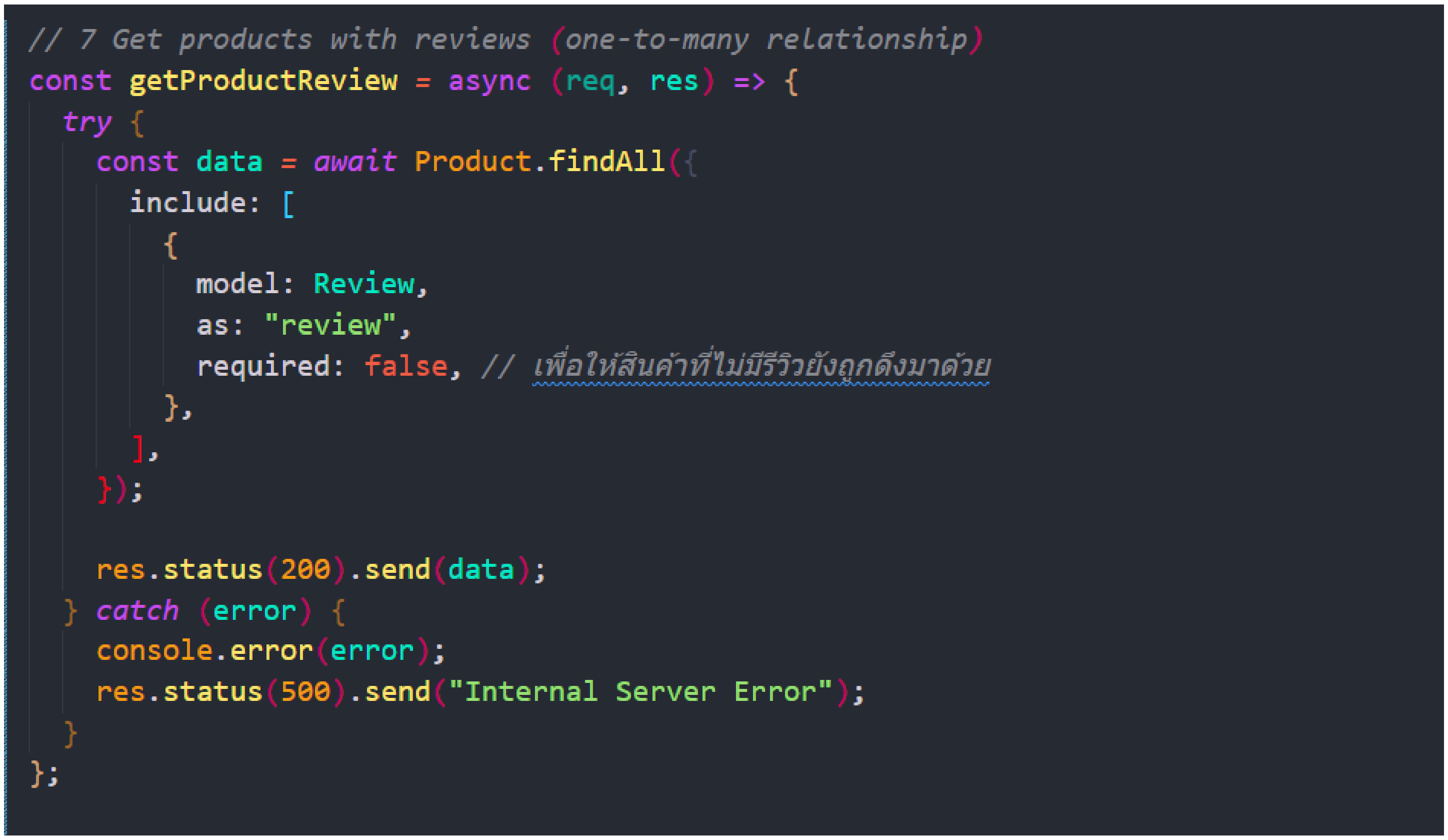
```
You, 3 minutes ago | 2 authors (You and others)
const productControllers = require("../controllers/productControllers.js");        You, 3 min
const reviewControllers = require("../controllers/reviewControllers.js");

// router
const router = require("express").Router();

// Product routes
router.post("/addProduct", productControllers.addProduct);
router.get("/getAllProducts", productControllers.getAllProducts);
router.get("/getPublishedProduct", productControllers.getPublishedProduct);
router.get("/:id", productControllers.getOneProduct);
router.put("/:id", productControllers.updateProduct);
router.delete("/:id", productControllers.deleteProduct);

// Review routes
router.post("/addReview", reviewControllers.addReview);
router.get("/getAllReview", reviewControllers.getAllReview);

```

13 edit index.js in models



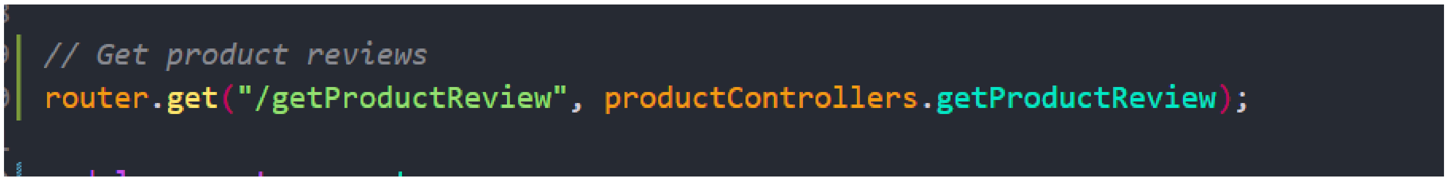
```
// กำหนดความสัมพันธ์ One-to-Many ของโมเดล "product" กับ "review"
db.product.hasMany(db.review, {
  foreignKey: "product_id",
  as: "review",
});

db.review.belongsTo(db.product, {
  foreignKey: "product_id",
  as: "product",
});

module.exports = db;

```

14 connect one to many



```
// 7 Get products with reviews (one-to-many relationship)
const getProductReview = async (req, res) => {
  try {
    const data = await Product.findAll({
      include: [
        {
          model: Review,
          as: "review",
          required: false, // เพื่อให้สินค้าที่ไม่มีรีวิวยังถูกดึงมาด้วย
        },
      ],
    });

    res.status(200).send(data);
  } catch (error) {
    console.error(error);
    res.status(500).send("Internal Server Error");
  }
};
```

15 create getProductReview in productRouter.js



```
// Get product reviews
router.get("/getProductReview", productControllers.getProductReview);
```