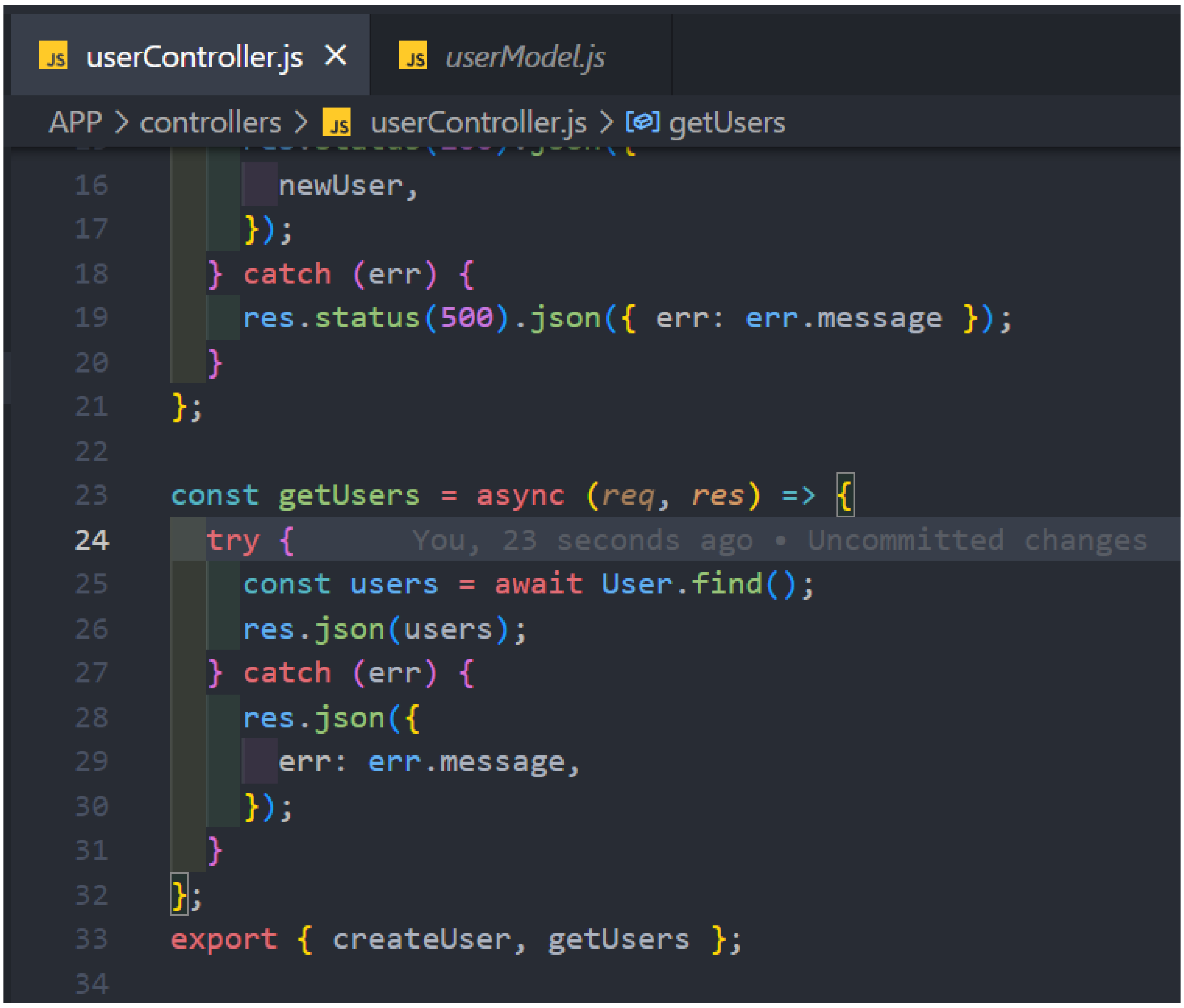
getAll users from db

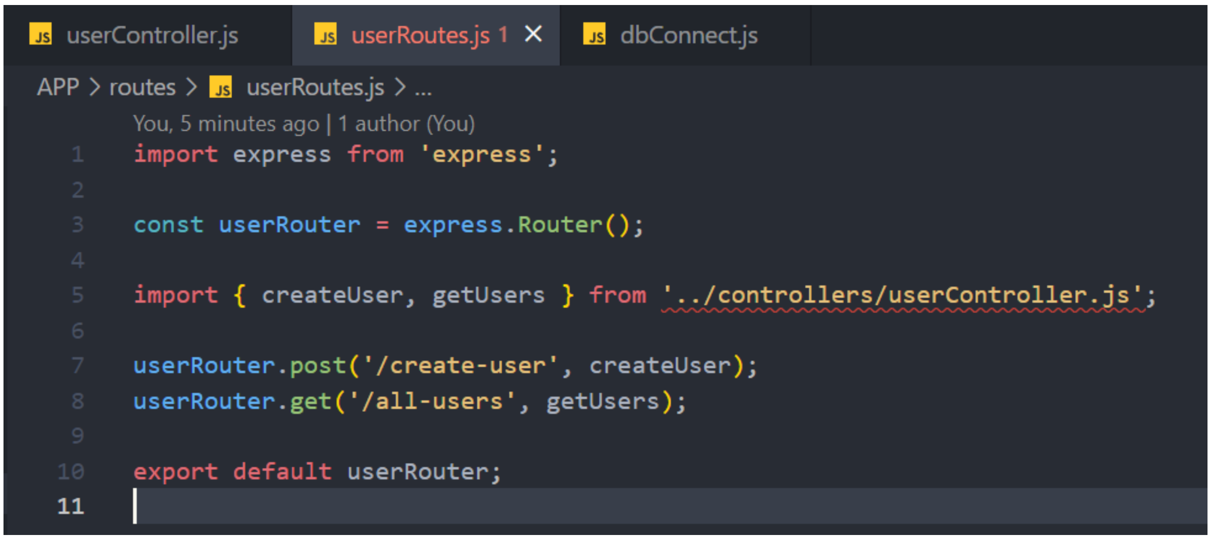


userController.js × userModel.js

APP > controllers > userController.js > getUsers

```
      newUser,
    });
  } catch (err) {
    res.status(500).json({ err: err.message });
  }
};

const getUsers = async (req, res) => {
  try {        You, 23 seconds ago • Uncommitted changes
    const users = await User.find();
    res.json(users);
  } catch (err) {
    res.json({
      err: err.message,
    });
  }
};
export { createUser, getUsers };
```

Add the Route



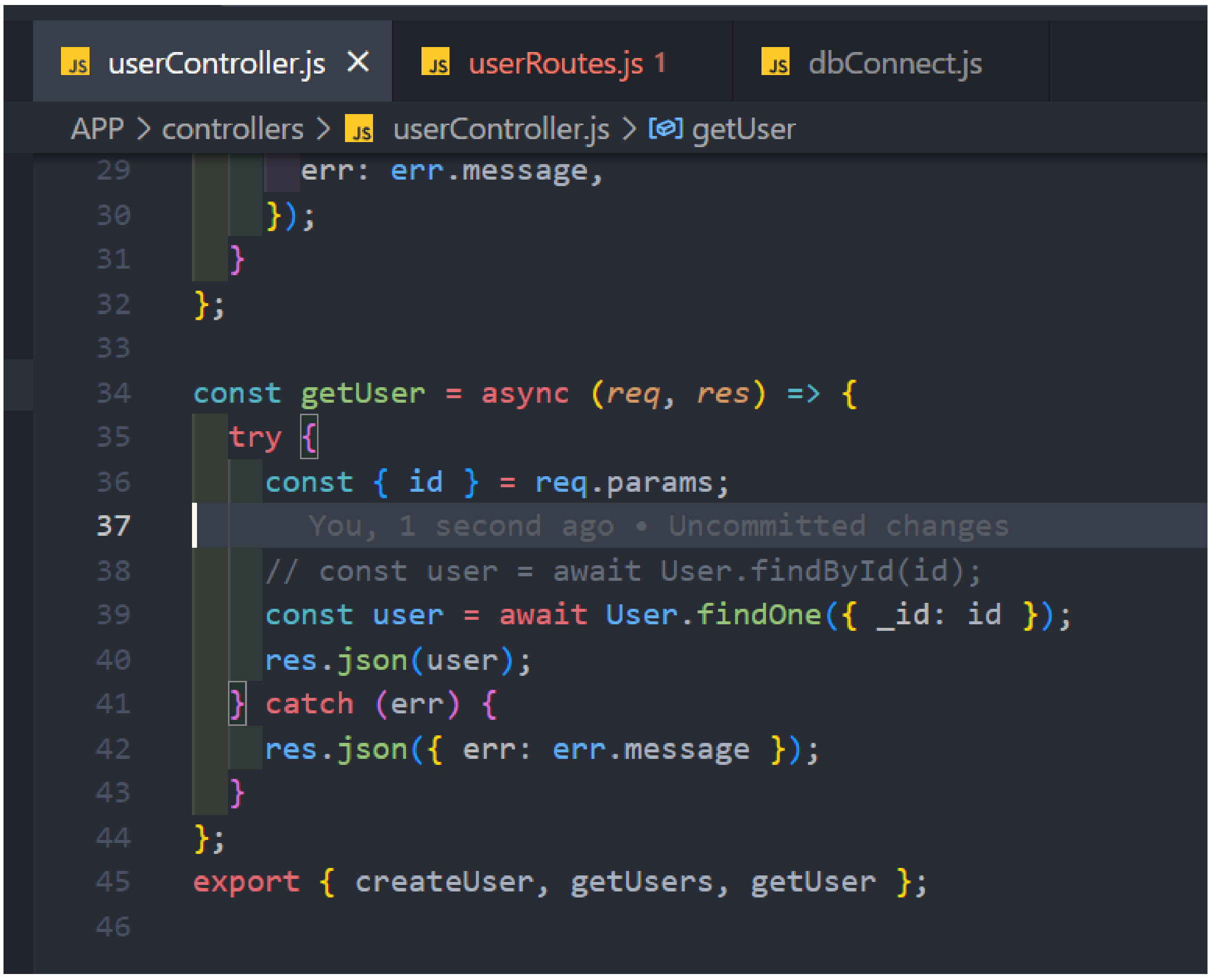
userController.js userRoutes.js 1 × dbConnect.js

APP > routes > userRoutes.js > ...

You, 5 minutes ago | 1 author (You)

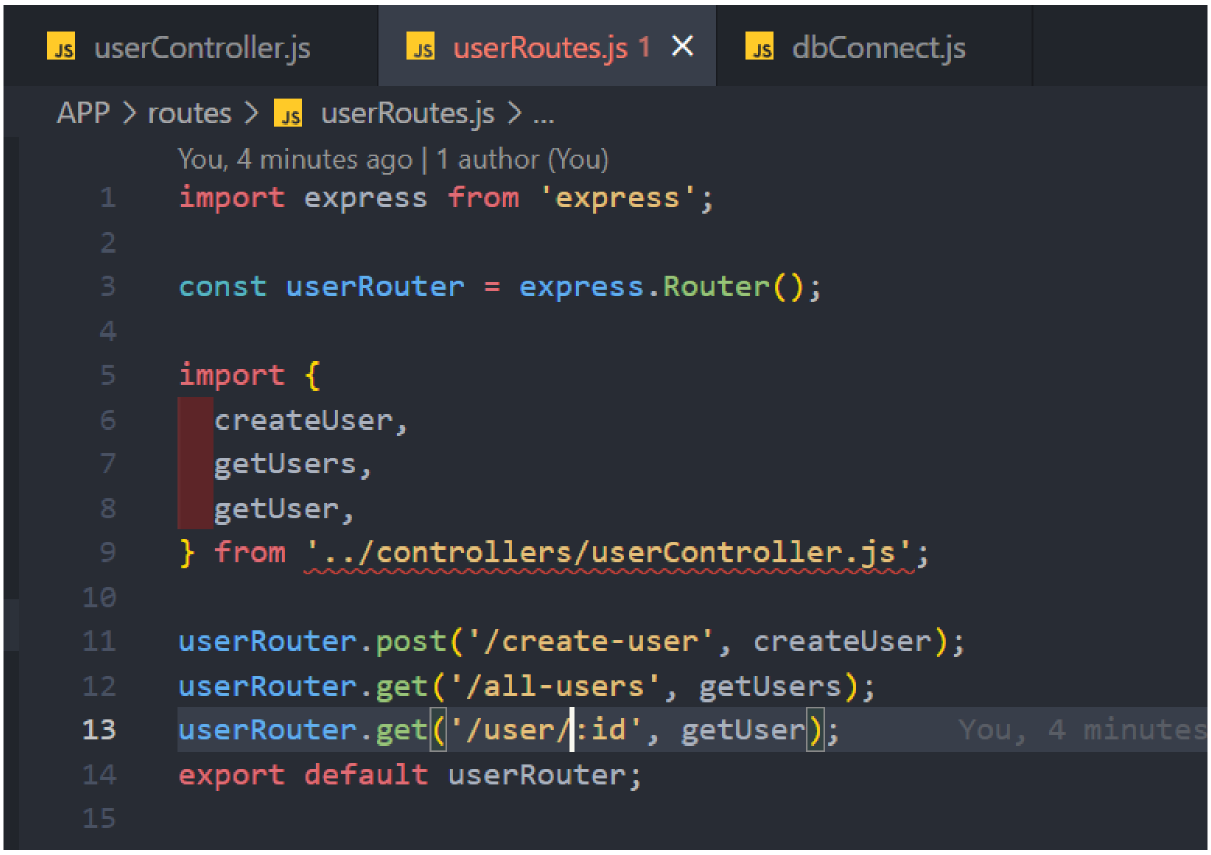
```
import express from 'express';

const userRouter = express.Router();

import { createUser, getUsers } from '../controllers/userController.js';

userRouter.post('/create-user', createUser);
userRouter.get('/all-users', getUsers);

export default userRouter;

```

getUser based on Id:



```
      err: err.message,
    });
  }
};

const getUser = async (req, res) => {
  try {
    const { id } = req.params;
      You, 1 second ago • Uncommitted changes
    // const user = await User.findById(id);
    const user = await User.findOne({ _id: id });
    res.json(user);
  } catch (err) {
    res.json({ err: err.message });
  }
};
export { createUser, getUsers, getUser };
```

Create route for getUser



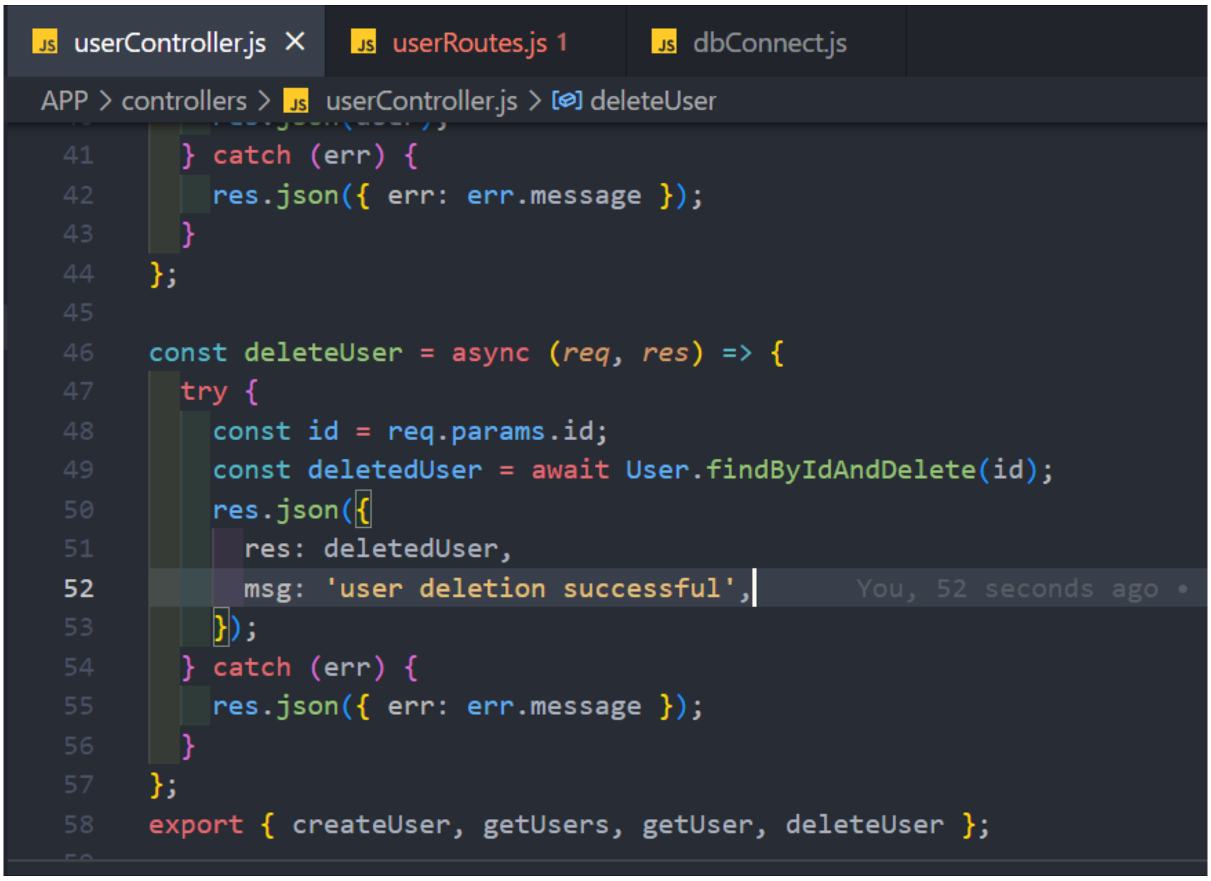
userController.js | userRoutes.js 1 | dbConnect.js

APP > routes > userRoutes.js > ...

You, 4 minutes ago | 1 author (You)

```
import express from 'express';

const userRouter = express.Router();

import {
  createUser,
  getUsers,
  getUser,
} from '../controllers/userController.js';

userRouter.post('/create-user', createUser);
userRouter.get('/all-users', getUsers);
userRouter.get('/user/:id', getUser);          You, 4 minutes
export default userRouter;
```

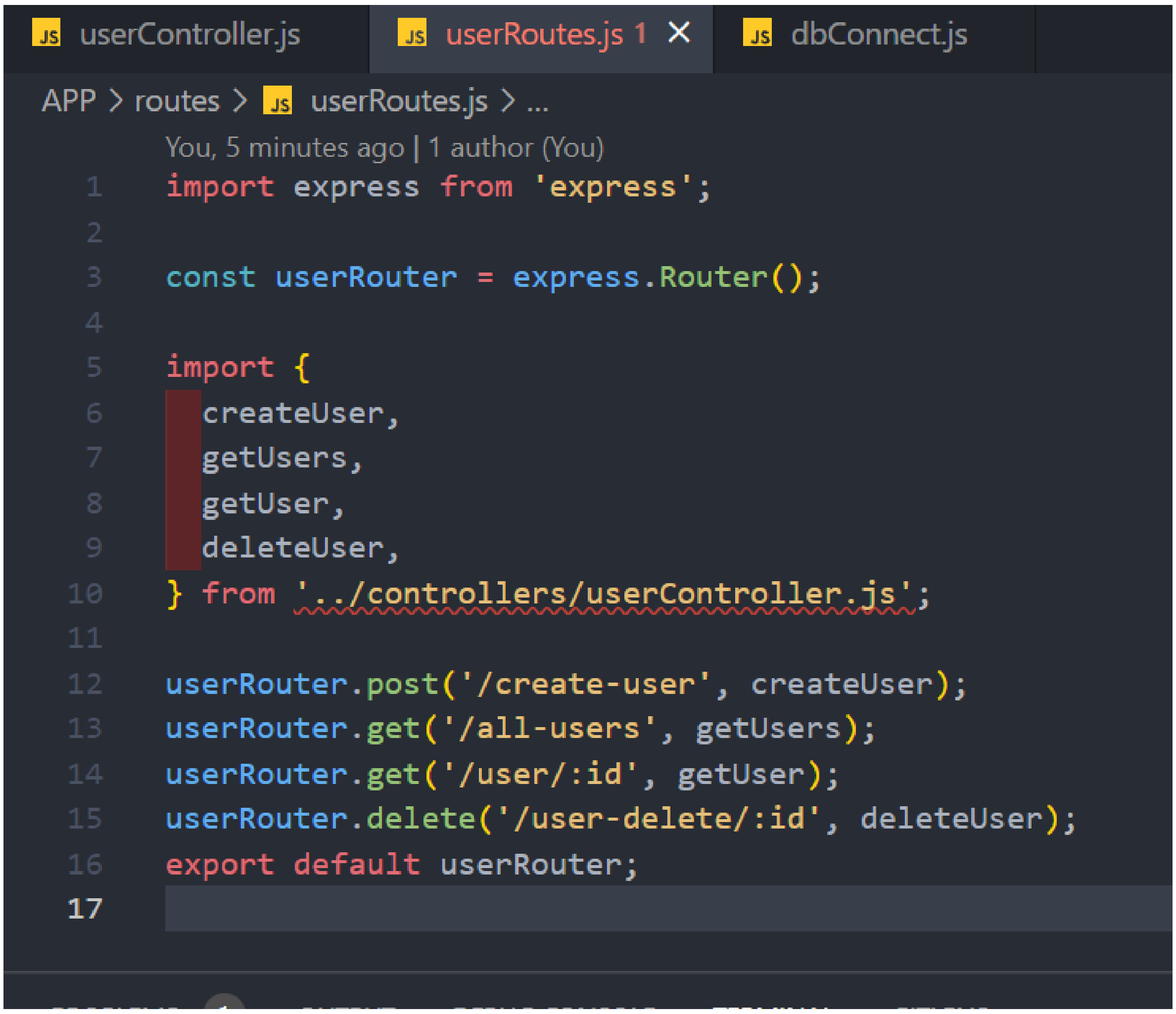
Deleteuser:



userController.js | userRoutes.js 1 | dbConnect.js

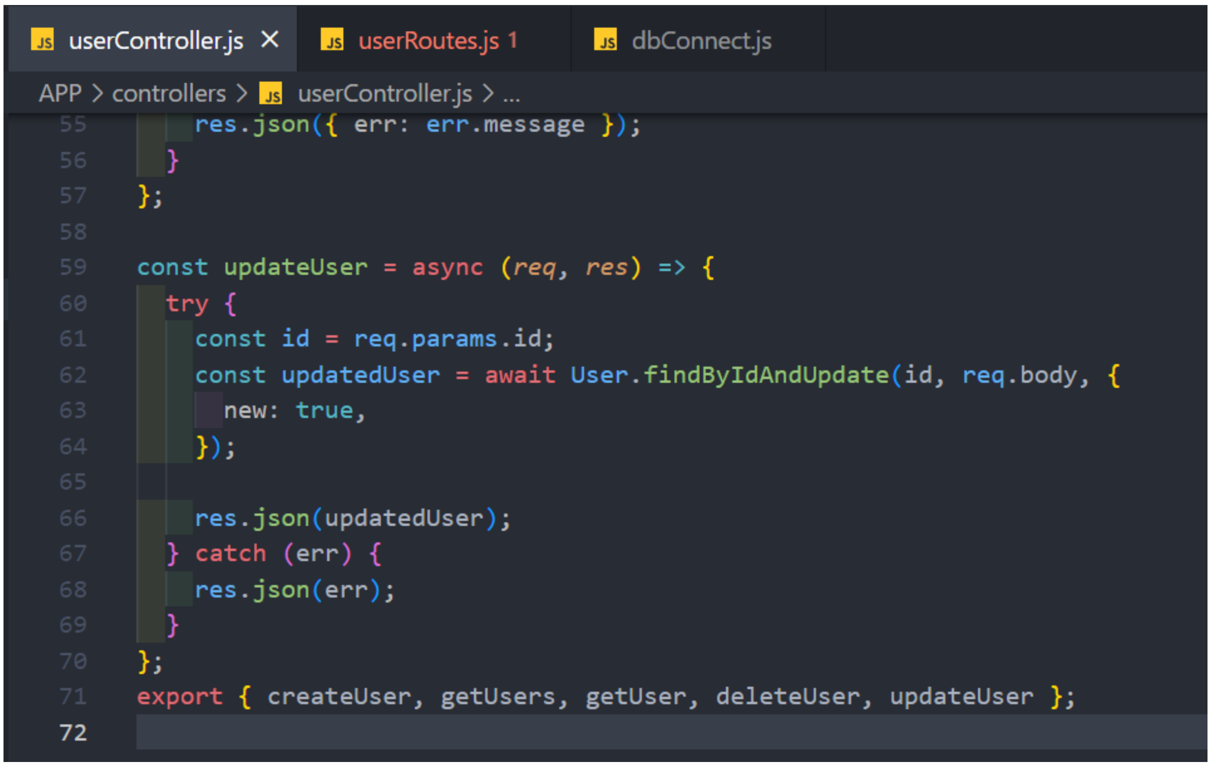
APP > controllers > userController.js > deleteUser

```
  } catch (err) {
    res.json({ err: err.message });
  }
};

const deleteUser = async (req, res) => {
  try {
    const id = req.params.id;
    const deletedUser = await User.findByIdAndDelete(id);
    res.json({
      res: deletedUser,
      msg: 'user deletion successful',        You, 52 seconds ago •
    });
  } catch (err) {
    res.json({ err: err.message });
  }
};
export { createUser, getUsers, getUser, deleteUser };
```

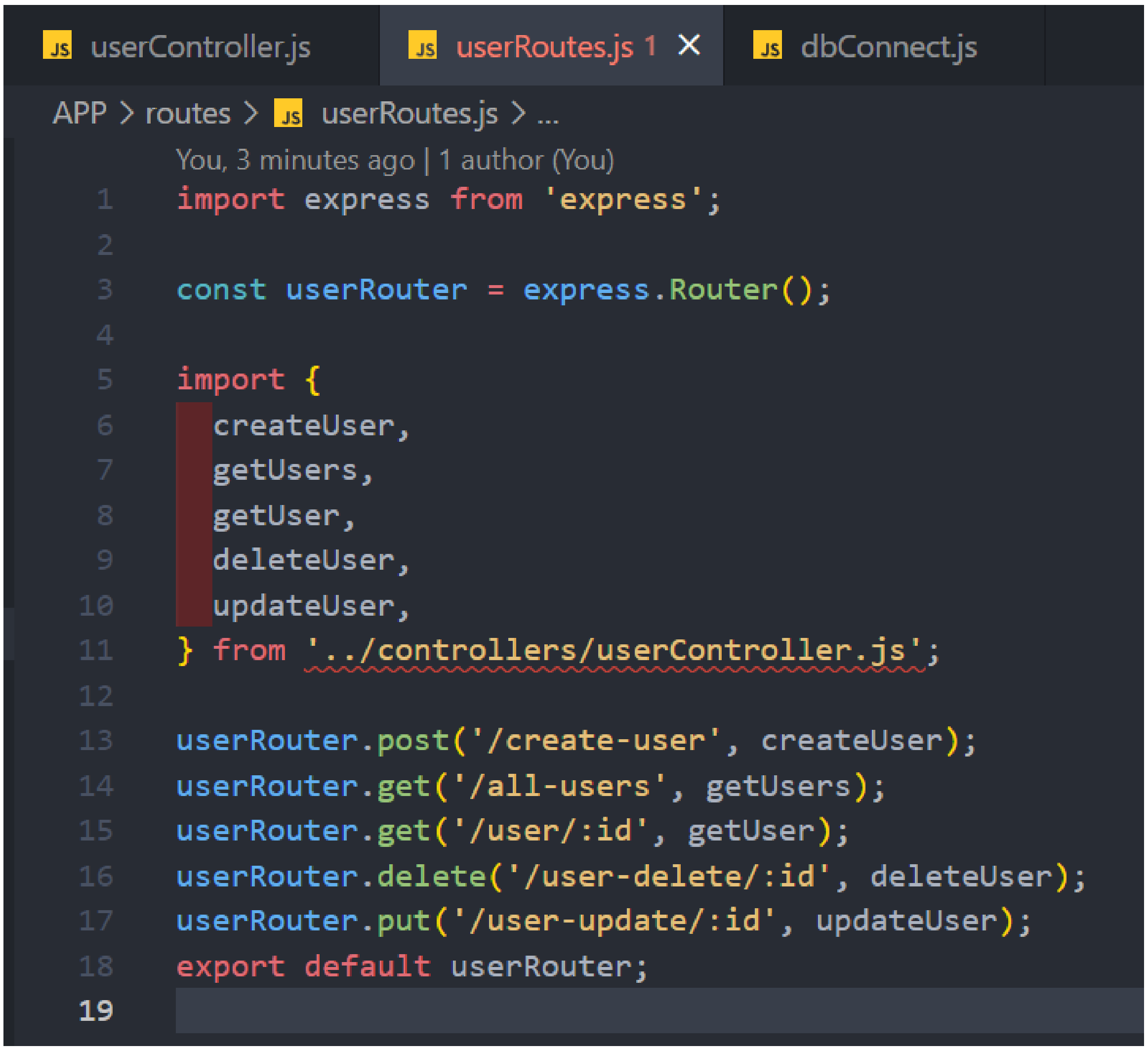

userController.js
userRoutes.js 1
dbConnect.js
APP > routes > userRoutes.js > ...
You, 5 minutes ago | 1 author (You)

```javascript
import express from 'express';

const userRouter = express.Router();

import {
  createUser,
  getUsers,
  getUser,
  deleteUser,
} from '../controllers/userController.js';

userRouter.post('/create-user', createUser);
userRouter.get('/all-users', getUsers);
userRouter.get('/user/:id', getUser);
userRouter.delete('/user-delete/:id', deleteUser);
export default userRouter;

```

Update User:



```javascript
    res.json({ err: err.message });
  }
};

const updateUser = async (req, res) => {
  try {
    const id = req.params.id;
    const updatedUser = await User.findByIdAndUpdate(id, req.body, {
      new: true,
    });

    res.json(updatedUser);
  } catch (err) {
    res.json(err);
  }
};
export { createUser, getUsers, getUser, deleteUser, updateUser };

```

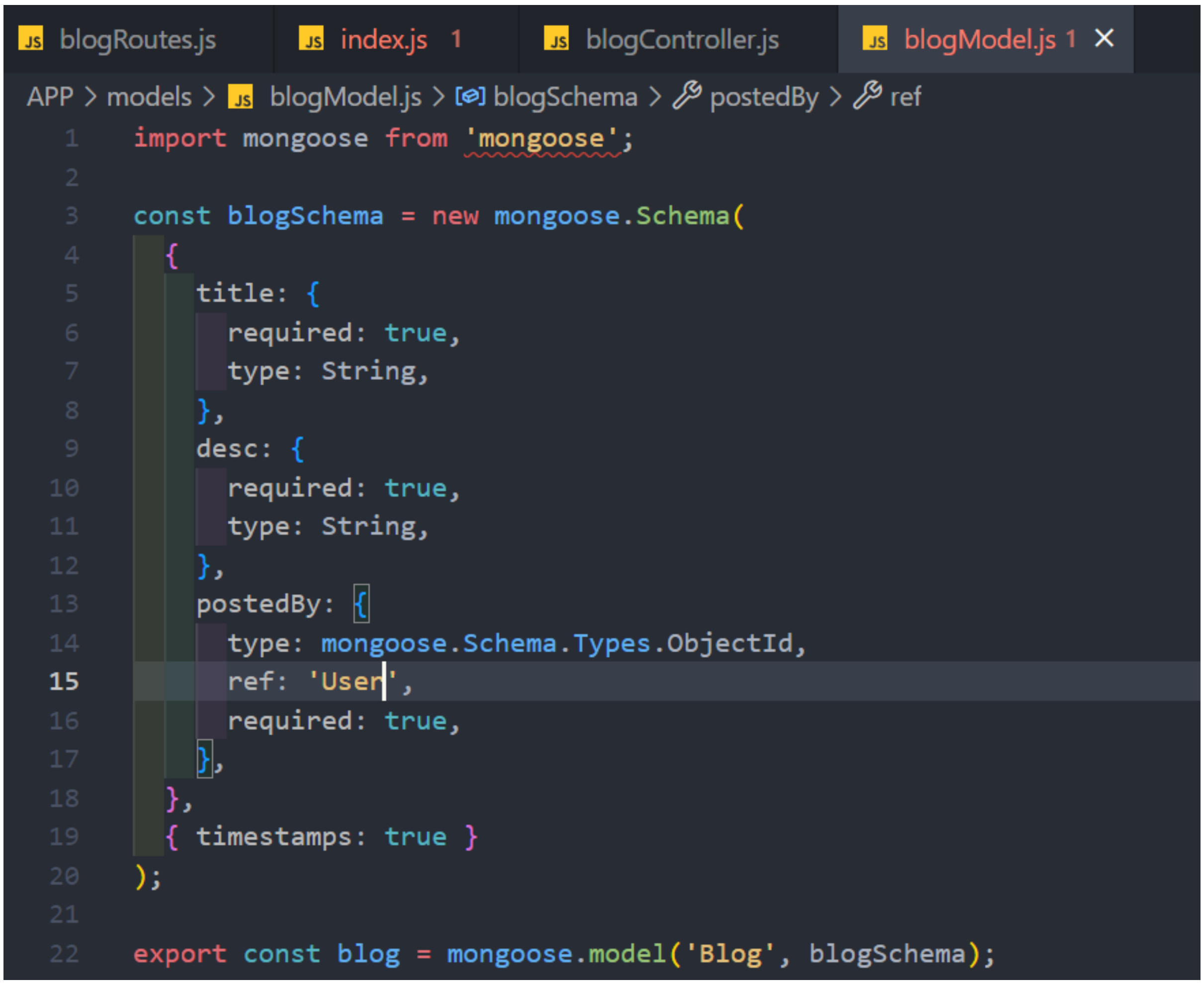

userController.js
userRoutes.js 1
dbConnect.js

APP > routes > userRoutes.js > ...

You, 3 minutes ago | 1 author (You)

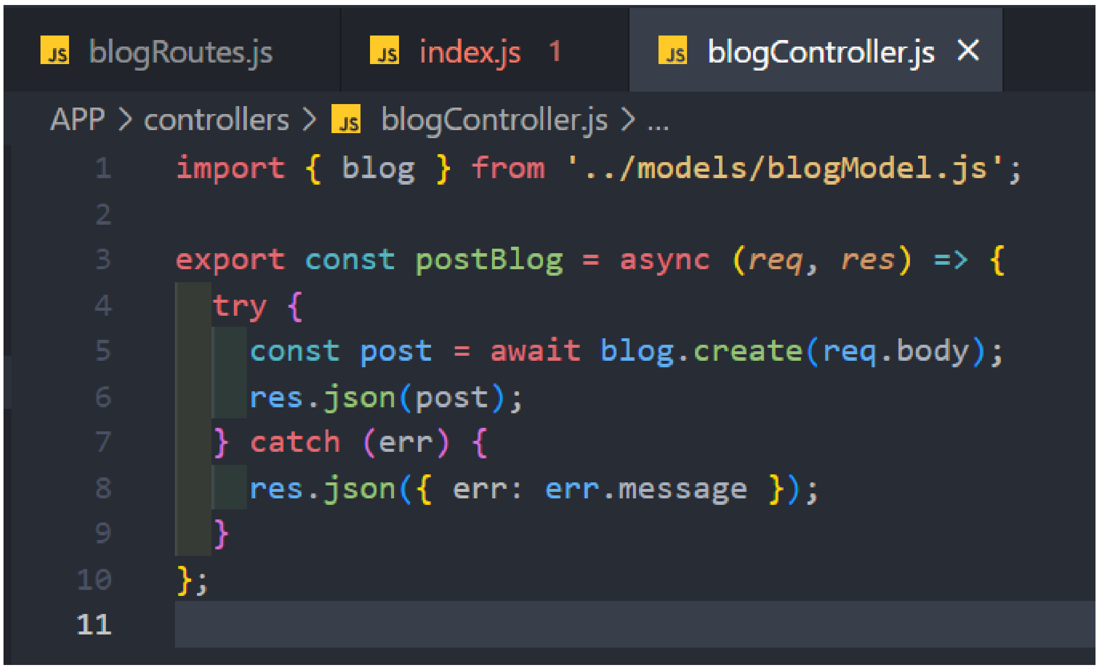
```javascript
import express from 'express';

const userRouter = express.Router();

import {
  createUser,
  getUsers,
  getUser,
  deleteUser,
  updateUser,
} from '../controllers/userController.js';

userRouter.post('/create-user', createUser);
userRouter.get('/all-users', getUsers);
userRouter.get('/user/:id', getUser);
userRouter.delete('/user-delete/:id', deleteUser);
userRouter.put('/user-update/:id', updateUser);
export default userRouter;

```

Create Blog Schema


blogRoutes.js | index.js 1 | blogController.js | blogModel.js 1

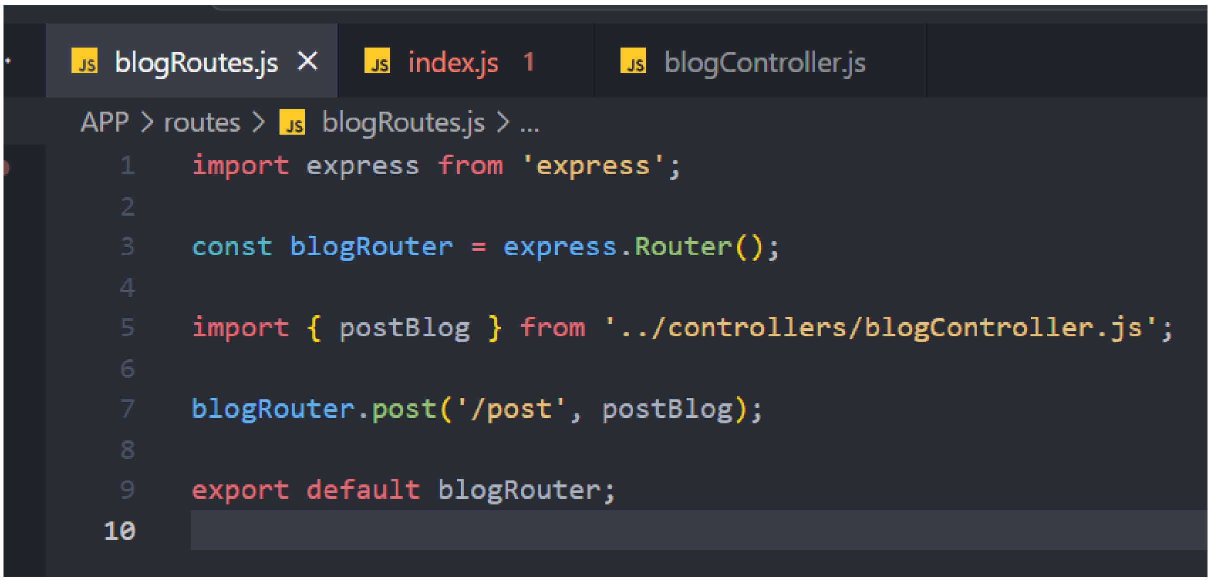
APP > models > blogModel.js > blogSchema > postedBy > ref

```
import mongoose from 'mongoose';

const blogSchema = new mongoose.Schema(
  {
    title: {
      required: true,
      type: String,
    },
    desc: {
      required: true,
      type: String,
    },
    postedBy: {
      type: mongoose.Schema.Types.ObjectId,
      ref: 'User',
      required: true,
    },
  },
  { timestamps: true }
);

export const blog = mongoose.model('Blog', blogSchema);
```

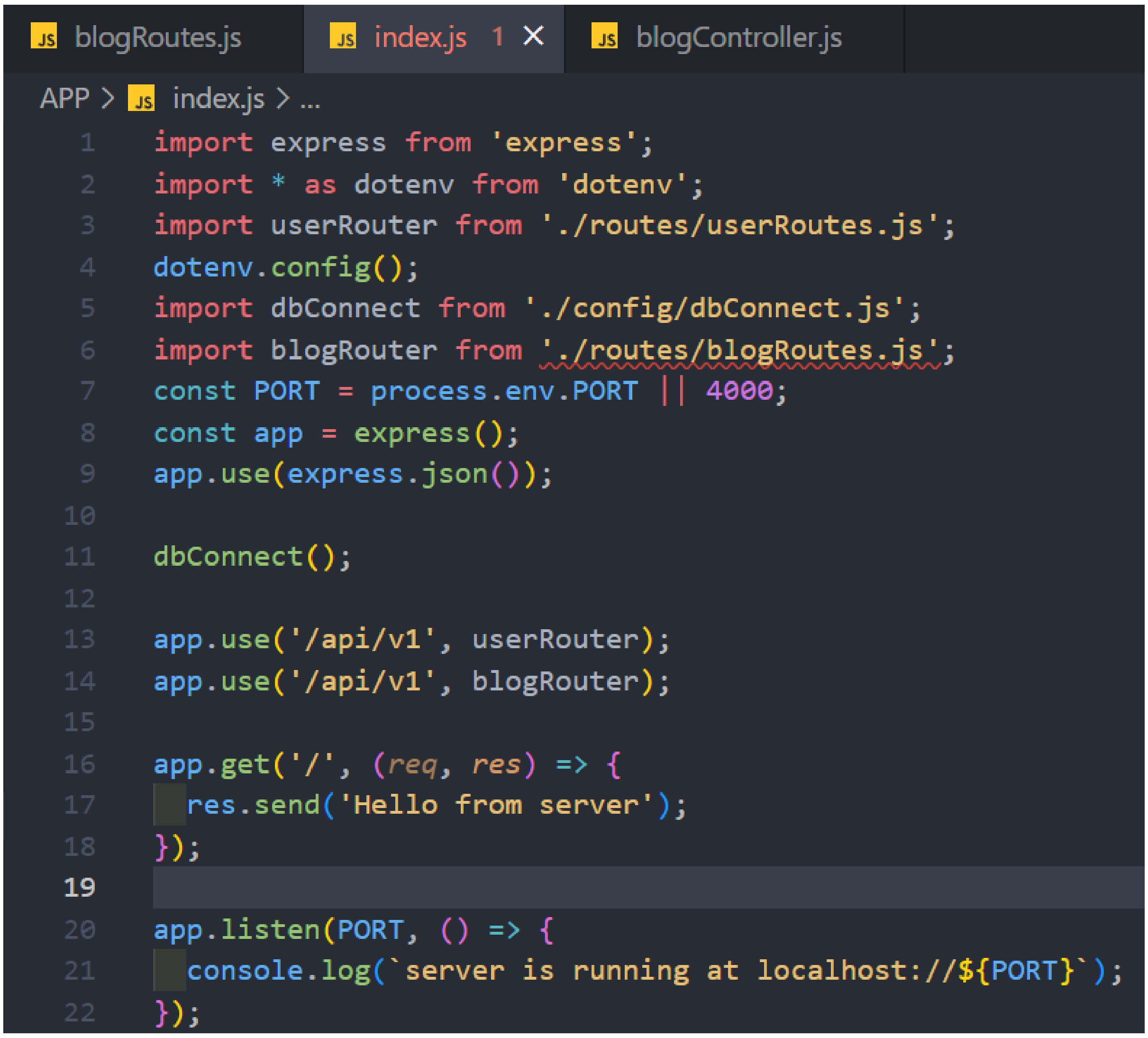

blogRoutes.js | index.js 1 | blogController.js

APP > controllers > blogController.js > ...

```
import { blog } from '../models/blogModel.js';

export const postBlog = async (req, res) => {
  try {
    const post = await blog.create(req.body);
    res.json(post);
  } catch (err) {
    res.json({ err: err.message });
  }
};

```

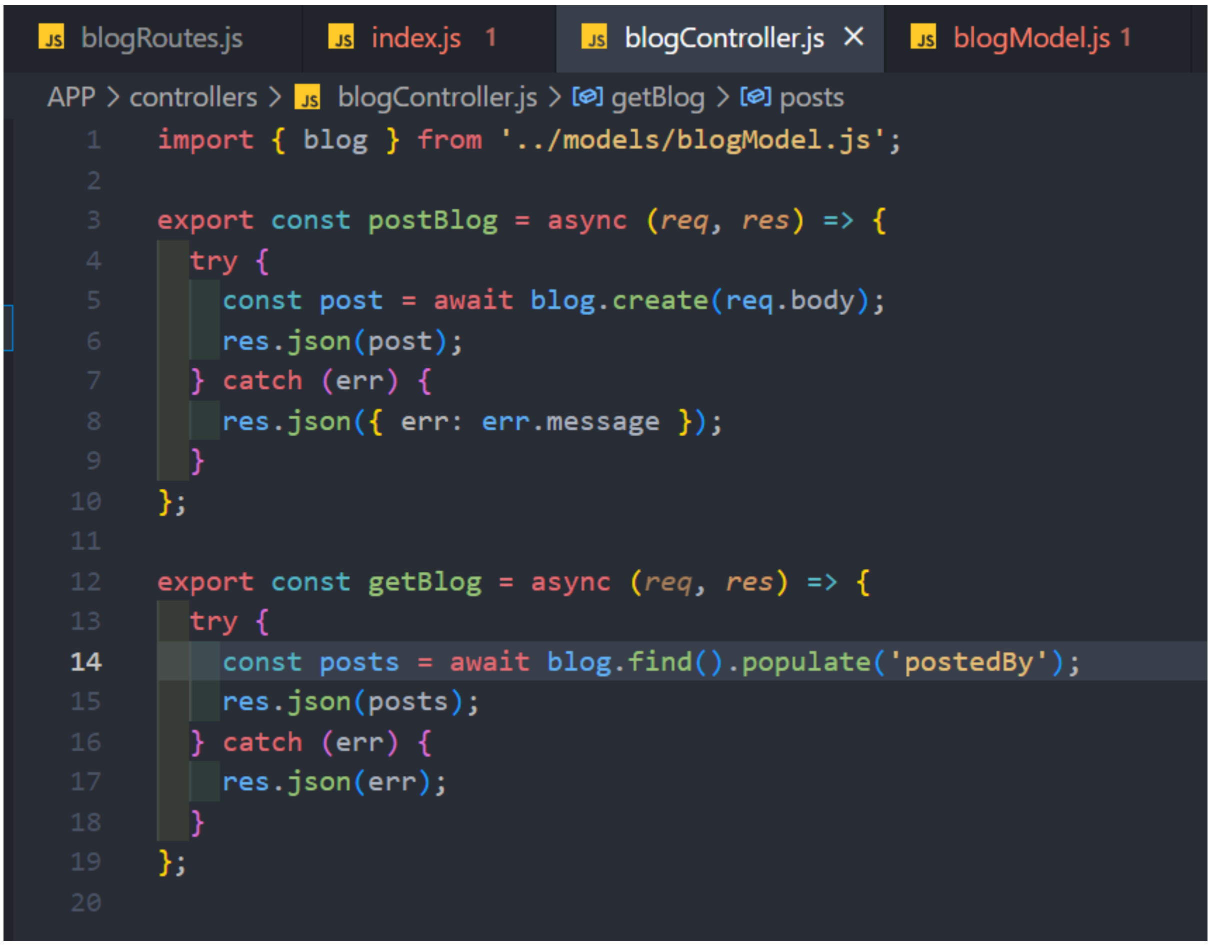


```javascript
import express from 'express';

const blogRouter = express.Router();

import { postBlog } from '../controllers/blogController.js';

blogRouter.post('/post', postBlog);

export default blogRouter;

```

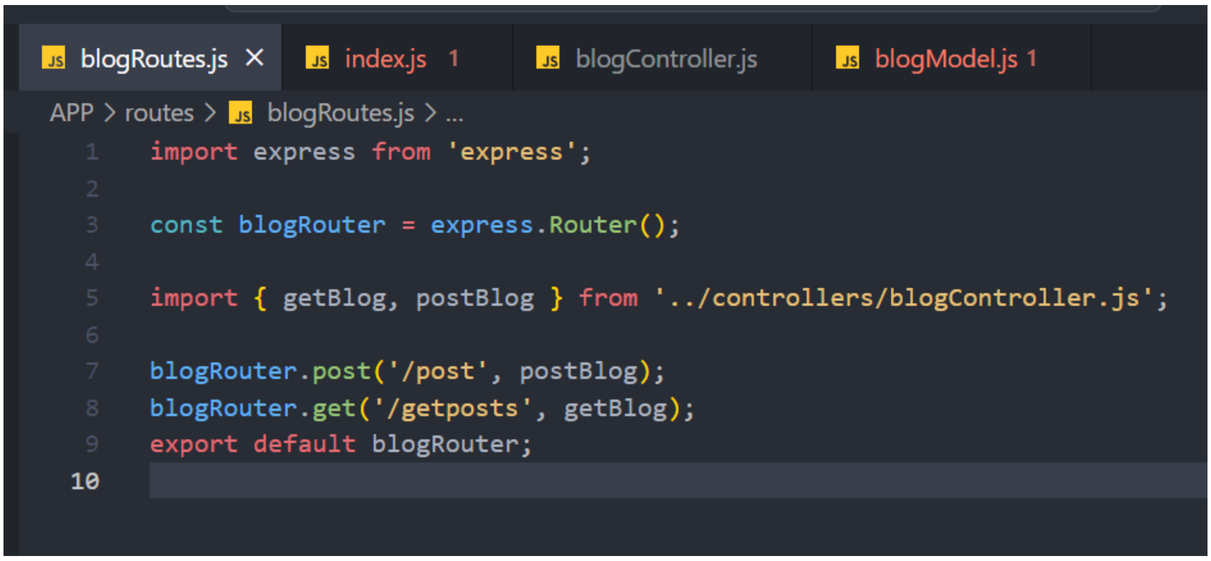


```javascript
import express from 'express';
import * as dotenv from 'dotenv';
import userRouter from './routes/userRoutes.js';
dotenv.config();
import dbConnect from './config/dbConnect.js';
import blogRouter from './routes/blogRoutes.js';
const PORT = process.env.PORT || 4000;
const app = express();
app.use(express.json());

dbConnect();

app.use('/api/v1', userRouter);
app.use('/api/v1', blogRouter);

app.get('/', (req, res) => {
    res.send('Hello from server');
});

app.listen(PORT, () => {
    console.log(`server is running at localhost://${PORT}`);
});
```

getBlog



```
import { blog } from '../models/blogModel.js';

export const postBlog = async (req, res) => {
  try {
    const post = await blog.create(req.body);
    res.json(post);
  } catch (err) {
    res.json({ err: err.message });
  }
};

export const getBlog = async (req, res) => {
  try {
    const posts = await blog.find().populate('postedBy');
    res.json(posts);
  } catch (err) {
    res.json(err);
  }
};
```



```
import express from 'express';

const blogRouter = express.Router();

import { getBlog, postBlog } from '../controllers/blogController.js';

blogRouter.post('/post', postBlog);
blogRouter.get('/getposts', getBlog);
export default blogRouter;
```

Read

Update

Delete

relation