

Node.js- Building a Modular Express.js Application with Router Factories

Creating modular, maintainable applications is a central tenet of good software design. In Node.js with Express, we can achieve this by implementing router factories. This approach not only keeps our code organized but also enables us to add reusable, configurable routes for various features in the app.

Setting Up a Modular Structure

Simple example

project structure

```
app/  
├─ server.js  
└─ users.js
```

Code Walkthrough

1. server.js

```
// app/server.js
const express = require('express');
const app = express();
const PORT = process.env.PORT || 1234;
const userMiddleware = require('./users');

app
  .use('/api/v1/', userMiddleware( {message:
    "User List..."}) );

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

2. users.js

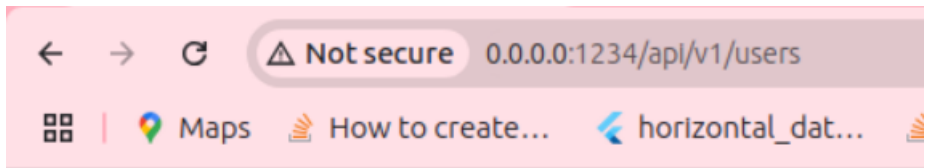
```
const express = require('express');
const users = function(options = {}){
  const router = express.Router();

  router.get('/users', (req, res, next) => {
    res.end(options.message);
  })

  return router;
}
module.exports = users;
```

Testing in browser:

```
http://0.0.0.0:1234/api/v1/users
```



User List...

A More Complex Example with a Service Class

project structure

```
app/  
├─ server.js  
└─ users.js
```

1. server.js

```
// app/server.js  
const express = require('express');  
const app = express();  
const PORT = process.env.PORT || 1234;  
  
const userMiddleware = require('./users');  
  
class UserService{  
  constructor(message = "Hello"){  
    this.message = message;  
  }  
  
  createMessage(name){  
    return `${this.message}, ${name}`  
  }  
}
```

```
}

app
  .use('/api/v1/service1', userMiddleware({
    service: new UserService('Hello, service
1')
  }))
  .use('/api/v1/service2', userMiddleware({
    service: new UserService('Hello, service
2')
  }));

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

2. users.js

```
const express = require('express');

const users = function(options = {}){
  const router = express.Router();

  // Get controller
  const {service} = options;

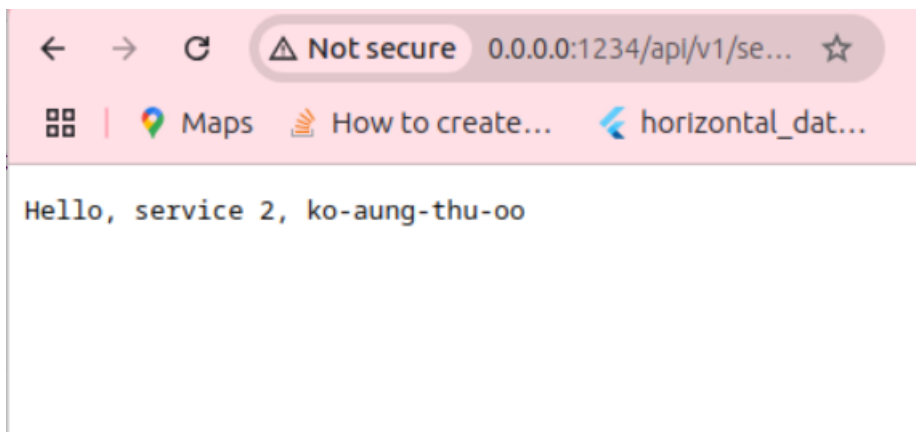
  router.get('/users', (req, res, next) => {
    res.end(
      service.createMessage(req.query.name ||
'Default message')
    );
  })

  return router;
}

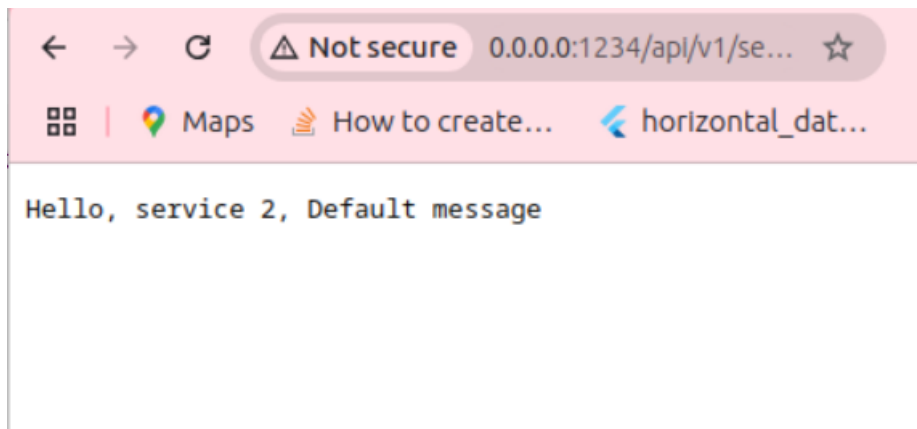
module.exports = users;
```

Testing in browser:

```
http://0.0.0.0:1234/api/v1/service1/users?  
name=ko-aung-thu-oo
```



```
http://0.0.0.0:1234/api/v1/service2/users
```

AUNG THU OO

I'm Seeking Job **OPPORTUNITIES**

Senior Developer Role

Bringing over 15 years of software industry experience, I use my full-stack skills to deliver scalable, user-friendly web and mobile applications that align with both user needs and business goals.



My Skills

- Next.js
- React.js
- Node.js
- Flutter
- Laravel

VISIT MY WEBSITE & EMAIL

**<https://aungthuoo.github.io>
koaungthuoo@gmail.com**