

Rutas

Ya aprendimos a hacer un servicio que se conecta con nuestra base de datos utilizando las distintas operaciones CRUD. Vimos que por cada ruta que creamos, en nuestro `index.js` creamos un método

`app.get()` | `app.post()` | `app.patch()` | `app.delete()`

Si quisiéramos seguir agregando más rutas nuestro `index.js` se volvería complejo de leer, por eso lo que se hace es separar las rutas y mantenerlas individuales.

Express Router

Vamos a crear lo que se llama **Router** y vamos a separar todas las rutas en el código para luego combinarlas y que quede nuestra aplicación ordenada.

Por convención, conviene tener un Router por cada recurso que tenemos:

- Productos
- Usuarios
- Etc.

Express nos permite crear un router instanciando una nueva clase del mismo:

```
const router = new express.Router();
```

Una vez que sesamos nuestro router, por cada ruta que creemos vamos a utilizar este mismo router de la siguiente manera:

```
router.get('/', (req, res) => { res.send('Home Page') });
```

Funciona similar al `app.get`, solo que esta vez estamos usando `router` y, para registrar el mismo y que funcione, nos falta una última línea que es decirle a nuestra aplicación que use este router que estamos creando

```
app.use(router);
```

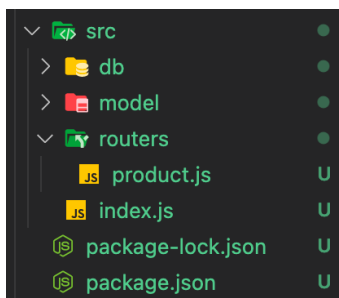
La lógica siempre va a ser (en este orden) :

1. Creo un router
2. Creo las rutas
3. Uso el router en mi app

Por ahora, esto no tiene mucho sentido porque las rutas siguen en un mismo lugar. Lo ideal es, como dijimos, separar las rutas en distintos archivos.

Separando las routes

Dentro de la carpeta src, nos vamos a crear una subcarpeta llamada routers y vamos a crear una ruta por cada recurso que tenemos, en nuestro caso solamente product.



Dentro de product.js, vamos a importar express, y vamos a crear un Router como hicimos en el punto anterior:

```
const express = require('express');
const router = new express.Router();
```

Ahora podemos crear todas las rutas que sean pertinentes al recurso de productos. Lo que vamos a hacer nosotros es migrar la información de la ruta que teníamos en el index.js, a este nuevo archivo.

A tener en cuenta:

1. Vamos a tener que - en el nuevo archivo - volver a importar el modelo de productos.
2. Vamos a tener que exportar nuestro router
3. Vamos a tener que usar nuestro router en el index.js

En el archivo router/product.js nos debería quedar algo de este estilo:

```
const express = require('express');
const router = new express.Router();
const Product = require('../model/product');
```

```
router.post('/product', (req, res) => {
```

```
  // Instancio un nuevo producto
  const product = new Product(req.body)
  product.save()
```

```
    .then(() => {  
      res  
        .status(201)  
        .send(product);  
    })  
    .catch( (err) => {  
      res  
        .status(400)  
        .send(err);  
    });  
  });  
});
```

```
module.exports = router;
```

Noten el cambio del **app.post** al **router.post**.

Finalmente, en el index.js vamos a tener que

1. Importar este router
2. Usarlo en nuestra app.

```
const productRouter = require('./routers/product');  
app.use(productRouter);
```

Y nuestro index.js queda mucho más ordenado.