TEMA 3 EXPRESS.JS

Parte II – Express aplicado a servicios REST

II.1 Modos de enrutamiento con Express

Enrutar con Express

- La estructura básica de Express:

```
const express = require('express');
let app = express();
app.listen(8080);
```

- Tenemos tres formas de enrutar:
  - Enrutar desde el servidor principal. Si la aplicación es muy simple
  - Enrutar desde enrutadores independientes. Si la aplicación es más compleja
  - Dividir los métodos del Servidor principal en módulos

1) Enrutar desde el Servidor Principal

- Se utiliza si tenemos pocos servicios a desarrollar.
- Definimos rutas en el propio servidor (app.js), añadiendo métodos (get, post, put o delete) indicando cada ruta que los debe atender y el callback que se debe ejecutar.

```
let app = express();

app.get('/bienvenida', (req, res) => {
res.send('Hola, bienvenido/a');
});
app.delete('/comentarios', (req, res) => {
// Código para borrar los comentarios...
});
...
```

2) Enrutar desde enrutadores independientes

- Aplicaciones complejas, con muchos servicios a atender o que queramos tratar de forma independiente los servicios relativos a conceptos distintos.
- Creamos distintos enrutadores por cada grupo de servicios deseado y luego mediante un middleware los vamos enlazando hasta el Server principal.
- Se suele crear una carpeta “ROUTES” y definir un archivo por cada grupo de rutas

Ejemplo

- Imaginemos que tenemos un conjunto de servicios destinados a la gestión de Usuarios con URI *usuarios/*
- El resto los englobaremos en la URI *principal/*
- Crearemos dos archivos por cada enrutador:
  - usuarios.js
  - index.js
- En ambos casos, utilizaremos un objeto Router que forma parte del objeto global que inicializamos con Express

USUARIOS.JS

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

let router = express.Router();

router.get('/', (req, res) => {
console.log("Listado de usuarios");
...
});

router.post('/', (req, res) => {
console.log("Inserción de usuario");
...
});
...

module.exports = router;
```

INDEX.JS

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

let router = express.Router();

router.get('/', (req, res) => {
console.log("Página de inicio");
...
});

router.get('/noticias', (req, res) => {
console.log("Listado de noticias");
...
});

...

module.exports = router;
```

APP.JS

- Incluimos los 2 enrutadores y los añadimos como middleware a la aplicación principal empleando USE:

```
const express = require('express');
const usuarios = require('./routes/usuarios');
const principal = require('./routes/index');

let app = express();
app.use('/usuarios', usuarios);
app.use('/', principal);

app.listen(8080);
```

Cuestiones importantes

- Observar los enrutadores son muy parecidos a los que hacíamos antes en el index.js principal
- El objeto router dispone de los métodos get, post, put o delete.
- Los enrutadores parten de la base que se indica en app.js, por lo que dentro en los enrutadores ya no hay que indicarlo

Añadir middleware en enrutadores concretos

- Podemos añadir *middleware* por separado también a cada enrutador, empleando el método use del propio enrutador.
- Por ejemplo, podemos añadir un middleware en el archivo "routes/usuarios.js" que muestre por consola la fecha actual:

```
let router = express.Router();
router.use((req, res, next) => {
console.log(new Date().toString());
next();
});
...
```

3) Dividir los métodos del servidor principal en módulos

- Es una combinación de los dos anteriores.
- En vez de usar enrutadores separemos los métodos de enrutamiento de la aplicación principal en métodos.
- Por ejemplo, si tenemos esta api Rest:

```
let app = express();

app.get('/', (req, res) => {
...
});

app.get('/noticias', (req, res) => {
...
});

app.get('/usuarios', (req, res) => {
...
});

app.post('/usuarios', (req, res) => {
...
});
```

Dividir en módulos II

- Creamos una carpeta para los diferentes módulos (controllers) y dentro 1 archivo por cada grupo de servicios.
- “controllers/usuarios.js” y “controllers/index.js”
- En ambos casos, lo que vamos a hacer es exportar una función que defina los distintos servicios.
- Dicha función recibirá como parámetro la aplicación sobre la que trabajar (el objeto app), para así poder definir los servicios sobre ella.

INDEX.JS y USUARIOS.JS

Index.js

```
module.exports = (app) => {
    app.get('/', (req, res) => {
    ...
    });
    app.get('/noticias', (req, res) => {
    ...
    });
}
```

Usuarios.js

```
module.exports = (app) => {
    app.get('/usuarios', (req, res) => {
    ...
    });
    app.post('/usuarios', (req, res) => {
    ...
    });
}
```

APP.JS

- Notar que ya no tenemos una ruta relativa a la base de cada enrutador, sino que debemos especificar la URI completa en cada servicio. La aplicación principal quedaría de este modo:

```
const express = require('express');
const usuarios = require('./controllers/usuarios');
const principal = require('./controllers/index');


let app = express();
usuarios(app);
principal(app);


app.listen(8080);
```