

## Rutas

En primer lugar, voy a hablar de rutas. Como vimos en clase, a través del sistema de ruteo de Express podemos definir, de manera sencilla cómo va a responder nuestra aplicación según el método HTTP y la ruta que está llegando al servidor. En nuestro caso usamos el método GET, el get lo que hace es una petición que es enviada la información por una URL.

En nuestro proyecto, las rutas se ubican en una carpeta de express llamada routes y desde acá se administra el sistema de ruteo de nuestro proyecto. A través de este sistema podemos definir como va a responder nuestra aplicación a un request según el método http y la ruta que esté llegando al servidor. Dentro de la carpeta routes, se encuentran los archivos js index, posts y users que administre los request a la ruta que tenga que ver con ese recurso. En este caso nuestros recursos son index, posts, users.

Las rutas definen qué controlador utiliza la response según la url y como bien explicaré Priscilla, los controladores son los encargados de la lógica del negocio. Para definir el sistema de ruteo es necesario requerir el módulo de express y guardar la ejecución de este método en una variable. Es por eso que todos nuestros archivos de rutas comienzan con `var express = require('express');`, porque de esta manera requiere el módulo de express. Sin embargo, ¿Qué me permite a mí hacer el método require? El método require me trae el módulo. ¿Qué módulo le voy a especificar dentro de los paréntesis? Bueno le voy a especificar un parámetro de un string (tipo de dato). Este string le voy a pasar el nombre del módulo. No le paso una ruta, sólo el nombre. A partir de esto, va a buscar a la carpeta “node-modules” si existe el módulo express, eso va a buscar su método principal de express y lo va a guardar en esta variable express.

Entonces esa variable, ¿qué va a contener? Esa variable va a ser un objeto literal que tiene métodos y tiene propiedades. Lo importante de esto es que estamos haciendo este require solo para utilizar el método router que nos provee express.

Luego sigue de otra variable llamada router, le ponemos operación de asignación, utilizamos el objeto literal que ya creamos anteriormente “express” y a express le doy punto y comienzo a visualizar todos los métodos y propiedades que tiene este objeto literal, de las cuales solo voy a necesitar el método “Router”. En router se va a guardar un objeto literal que me va a permitir a mí manejar y manipular las rutas de mi aplicación. ¿Qué más me va a permitir esto? Me va a permitir modularizar, que yo guarde en esta variable mi sistema de ruteo que pertenezca a este recurso.

Por último, requerimos dentro de una variable el controlador al que le queremos asignar a la ruta.

Para definir las rutas que consideramos necesarias para manejar distintos tipos de request. Ponemos “router” y punto. El punto nos indica todos los métodos que tiene este objeto literal. Hay varios métodos HTTP. Solo trabajamos con el método Get. Solo vamos a pedir información al servidor. Luego sigue por el path, parámetro de tipo string. Aca le voy a especificar que esta funcionalidad que voy a hacer a continuación va a ser ejecutada solo si entra en esta ruta. “/” que si no me equivoco se

le dice sufijo, y como segundo parámetro recibe el nombre de la variable en donde requerimos el controlador seguido del método que le hemos puesto a este. Por último, los archivos de las rutas terminan con un `module.exports = routes` para que el módulo pueda ser utilizado luego en `app.js`. ¿Por qué la variable “ROUTER”? Para que se exporte con todas las modificaciones.

Ya requerimos ese módulo pero, ¿Qué hacemos con ese objeto literal que nos trajo? Dentro de `app.js` hacemos que el método “use” reciba dos parámetros. Nos permite configurar métodos y propiedades para que mi aplicación levante de manera correcta.

El primer parámetro de tipo string es el nombre del prefijo. El prefijo que va manejar el nombre del recurso.

¿Qué es prefijo? Es el nombre del recurso. Si el recurso es autos, el prefijo es “/autos”. Si es libros es “/libros”

¿Cuál es el sufijo? Dentro del archivo rutas, ahí se va manejar los sufijos. Son funcionalidades del recurso.

Como segundo parámetro es el nombre de la variable que almacenamos anteriormente. “indexRouter”.

Por último, existen las rutas parametrizadas que son aquellas que definen qué parámetro va a ir cambiando, una parte de la ruta es fija y la otra variable. El parámetro variable o optativo se pone utilizando los `:`. Esto fue utilizado la ruta de post y user para hablar de el detalle de los post y según el post que el usuario clickee, va a variar el número de id. Después sigue el parámetro opcional `?` que declara que el id específico puede llegar o no a través de la url. `localhost` nos tira el id específico.

#### TENER EN CUENTA:

Router es un objeto literal que nos permite modularizar nuestro sistema de ruteo. Ahora “.get” es el método HTTP que espero que reconozca esta ruta. Este método recibe dos parámetros, el primer parámetro es la ruta que quiero que escuche, esta ruta es un sufijo. LOS PREFIJOS VAN EN EL APP.JS. Luego, como segundo parámetro tenemos un callback. Un callback es una función que pasa como parámetro de otra función. Este callback recibe 2 parámetros “req,res”. Ambos son objetos literales. Req me trae información de la solicitud, y res me trae información de la respuesta que le quiero dar al usuario. Luego está el cuerpo.