

Ahora vamos a proceder a limpiar un poco nuestro archivo de entrada, el archivo principal en nuestro, app.js, vamos a crear rutas específicas, vamos a habilitar la lectura de datos del formulario y los recogeremos en el back end, esto es algo muy interesante muy importante, es vital que aprendamos a hacerlo, cómo recoger los datos que envía un usuario a través de un formulario y tomarlos en el back-end para luego hacer algo con ellos. Para completar la tarea de enviar nuestros mails a través de este formulario primero tenemos que hacer algunas cosas en nuestro proyecto y además debemos capturar los datos que el usuario ingrese en estos campos en el backend, eso es lo primero que tenemos que hacer por lo tanto vamos a ver cómo está nuestro proyecto en este momento. Ok, aquí tenemos el servidor corriendo, por lo tanto vamos a comenzar a trabajar en nuestro archivo app.js aquí tenemos que indicar de alguna manera a express que vamos a habilitar la lectura de los datos que proceden de un formulario, es decir, de cualquier formulario que tengamos en nuestro proyecto, para eso vamos a utilizar un middleware, es un middleware nativo de express que permite la lectura de datos enviados desde un formulario, ok, simplemente diremos a app, views y vamos a usar dijimos que es un método nativo de express por lo tanto invocamos primero express y luego al método url o url en coudet, ok, perfecto, aquí tenemos entonces nuestro lector de datos de formulario, aquí podríamos agregar un parámetro es opcional pero bueno suele escribirse el extended en false, este es un dato secundario pueden buscarlo en la documentación lo que hace la diferencia entre que éste en true o en false es, bueno, qué librería va a utilizar para interpretar los datos, digamos, un tema menor pero bueno, lo ponemos porque ya es costumbre, ok, una vez que tenemos entonces habilitada la posibilidad de que nuestra aplicación reciba los datos del formulario vamos a ver cómo viajan los datos a través de un formulario, aquí tenemos nuestro formulario de contacto, tenemos que indicar dos cosas aquí muy importantes, primera, la primera es que archivo en este caso qué ruta verdad, va a controlar lo que ocurra con este formulario, es decir, qué ruta qué controlador va a recibir los datos de nuestro formulario, vamos a indicar que la ruta es bueno la tenemos aquí arriba en rutas dónde están nuestras rutas, caramba, vamos a indicarlo, aja bien nos falta ese paso todavía, bueno, entonces vamos a hacerlo, vamos a ir por ese paso que es un paso primero y tiene que ver con la limpieza de nuestro archivo, de paso miren vamos a ir mejorando algunas cuestiones, esta línea que acabo de ver aquí pues sinceramente sobra es una línea duplicada la que está aquí arriba hace exactamente lo mismo así que no tiene sentido que la dejemos, bien, les hablaba de rutas pero claro todavía no las hemos creado, las rutas o mejor dicho en la jerga de node, los routers o fábricas de rutas son carpetas con archivos que nos sirven para especificar qué archivo va a manejar cada una de las rutas e ir limpiando la estructura de nuestro archivo principal, para hacer esto lo primero que tenemos que hacer es bueno crear una carpeta por ejemplo que se llame rutas lo haremos aquí router, ok, y aquí dentro vamos a comenzar por indicar qué archivos manejarán las rutas que ya tenemos definidas, miren tenemos la ruta, esta podría ser la ruta index verdad, porque es la ruta principal o home y esta es la ruta contacto, crearemos sendos archivos aquí dentro con esos nombres haremos uno index.js y haremos un contacto.js, muy bien y aquí dentro vamos a mudar las funciones que tenemos aquí, vamos a llevarnos esto de aquí de index y lo vamos a poner aquí adentro y luego haremos lo siguiente, es una estructura bastante repetitiva así que una vez que digamos le toman la mano ya después es bastante sencillo, vamos a obviamente a traer express porque vamos a necesitar una de sus funcionalidades, en este caso

el router, el enrutador de express es lo que vamos a necesitar y aquí vamos a crear una constante que se llame, bueno, se puede llamar router, enrutador, como ustedes quieran, es bastante común ponerle router y aquí vamos a asignarle la función les dije router, el método router de express, ok, y ahora aquí ya no usaremos más el get simplemente diremos router, siempre haremos lo mismo, finalmente para que este módulo, este router esté disponible aquí, de donde lo hemos sacado vamos a exportar este módulo con la sintaxis `model.export` y el nombre del módulo que queremos exportar, bueno, cuál es la función en este caso router, perfecto, ya lo tenemos listo, vamos a hacer algo muy similar, es más vamos a copiar y pegar en contacto y aquí en contacto simplemente vamos a cambiar los nombres verdad, aquí la ruta será, bueno, no tenemos que cambiar nada la verdad solamente aquí, que render vamos a hacer, no vamos a renderizar la ruta index sino que miren, aquí en app renderizamos la vista contacto verdad, por lo tanto esa será la que renderizaremos, aprovechamos para borrar esta de fotos que habíamos puesto solamente en los momentos iniciales de las clases de node para ir probando, ven que este archivo va quedando mucho más pequeño, volvemos a contacto y aquí decimos que vamos a renderizar la vista contacto, recuerden el render corresponde a las vistas, tenemos una index y una contacto, ok, ahora tenemos que requerir, importar esas esas manejadores de rutas para poder utilizarlos aquí porque el punto de entrada de nuestras de nuestra aplicación es este archivo, aquí es donde vienen las peticiones, las request de los usuarios, por lo tanto aquí vamos a importar, podríamos hacerlo aquí por ejemplo, vamos a importar nuestros routers, lo haremos una vez por cada uno, tenemos uno para el index así que le podríamos poner index router como nombre y esto va a requerir el manejador que acabamos de crear que está en un archivo en la carpeta ./, tengo deshabilitada el teclado numérico, ahora si, ./routers/index, aquí está, muy bien, no hace falta escribir .js pues estamos en el entorno de node y node interpreta que queremos trabajar con esos archivos, voy a copiar esta línea y vamos a traernos también el manejador de la ruta contacto, vamos a ponerle de nombre contacto router y aquí el archivo que queremos traer de aquí de las rutas es este, contacto, porque allí es donde ahora delegamos el manejo, la lógica para trabajar con nuestras rutas, aquí lo único que va a pasar es lo siguiente, nuestra aplicación va a hacer vamos a decir use y cuando entra a una petición de la ruta raíz coma ya sabemos que eso lo maneja este enrutador, este controlador, el index, que aquí le hemos llamado como index router, bien, vamos a ponerlo entonces index router y cuando entre una petición de la ruta contacto, bueno, será manejada por el controlador que está en el controlador llamado contacto, aquí dentro de rauch, fíjense que aquí lo importamos y le pusimos de nombre contacto ruta, ok, vamos a ver ahora que nos está faltando para que esto corra, vamos a ver si lo exportamos, ok, si tenemos explotado y aquí también, revisemos los nombres por las dudas, ok, solo para chequear que no haya algún error verdad, perdón aquí tenemos un problema he confundido los nombres aquí en realidad hemos cambiado app por routers y no por el nombre del método, nos está faltando el método si, aquí es routers.get y aquí es router.get también, ahora sí, perfecto, ahora va a funcionar correctamente, debería estar todo igual que antes y ustedes me dirán bueno y para qué lo hacemos, ya lo verán que cuando nuestro proyecto vaya creciendo será mucho más cómodo tener separadas las rutas fuera de nuestro archivo principal, porque esta ruta luego van a ir creciendo, estos controladores, aquí habrá mucha más lógica, ok, muy bien, entonces ya tenemos las rutas separadas, ahora vamos a ver qué pasa con nuestro formulario, nuestro formulario aquí de contacto va a enviar estos datos a este controlador,

ahora sí lo hemos creado, entonces aquí tenemos que indicar su nombre/contacto, bien, porque /contacto porque la petición entra aquí a nuestro app y va a venir si viene simplemente a barra a la raíz tenemos que ir al index router, pero si viene así, es decir, si fueron al formulario de contacto noten que esta es la ruta, aquí está entrando y de aquí lo enviamos al controlador, entienden, y esa es la vía, primero entra aquí y luego va al controlador, por eso aquí en el formulario indicamos /contacto, muy bien, aquí dentro ahora vamos hacer lo siguiente, vamos a hacer un manejador, ya tenemos un manejador que nos muestra el formulario, ahora tenemos que hacer uno que reciba lo que el usuario envía del formulario, para indicar que el usuario va a enviar algo aquí ya dijimos que archivo va a procesar el envío pero tenemos que indicar el método, va a ser un método de tipo post, esto se refiere a los métodos http, cuando los métodos son de tipo get no hace falta escribirlos aquí porque es el método por defecto, el método get muestra el formulario pero una vez que el formulario está mostrado y el usuario rellena los cambios, los campos y le dan click y se dispara este botón, el formulario con todos sus datos cargados va a ser enviado a este archivo con el método post, este archivo es el mismo que muestra el formulario pero va a entrar no con método get ahora cargado de datos va a entrar con un método post, por lo tanto debemos hacer un controlador para esa situación que básicamente aquí hacemos exactamente igual que arriba verdad, esto es exactamente lo mismo, muy bien, pero ahora vamos a ver y dejaremos aquí la clase cómo podemos capturar los datos, los datos que el formulario nos envía aquí vienen en este objeto, en el objeto request, recuerden el objeto response es el que utilizamos nosotros como programadores para enviar una respuesta y el objeto request abreviado aquí como req o petición es en el cual viajan los datos o las peticiones de los usuarios, por lo tanto si aquí agregamos un simple console-log de la request particularmente del cuerpo de la request, ahí es donde vienen estos datos, vamos a ver aquí abajo nada más en esta primera instancia, los datos que nos envía el usuario, si logramos eso, bueno, ya podremos pasar a la siguiente clase vamos a ver si lo conseguimos, aquí tenemos nuestra página, ok, vamos a actualizar sólo por las dudas y el usuario aquí va a enviar algo, algunos datos y Antonio, el email es Anthony, bueno cualquier cosa no importa, es un mail cualquiera y aquí un mensaje que puede ser lo que sea, cuando le damos a enviar si todo funciona bien debería entrar aquí a este controlador la información y nosotros la veríamos en pantalla, listo, ya tenemos una buena parte de nuestro trabajo resuelta, cómo capturar los datos desde el front end y atajarlos recogerlos en el backend, en la próxima clase enviaremos esos datos a un servidor de correo.