Models

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

definimos el modelo de “country”, destructuramos de sequelize DataTypes para usarlo y especificar el tipo de valores que permitirá en ese campo.

Usamos constrains(restricciones) como allowNull: false

Texto

Descripción generada automáticamente

Definir modelo(representación de una tabla) llamado “activity” que luego se pluralizara(debió a una librería) en la BD, con los campos(atributos) de nombre, dificultad, duración y temporada. El id lo genera automáticamente. Exportamos dicho modelo. Permite valores vacíos ya que no es un campo obligatorio.

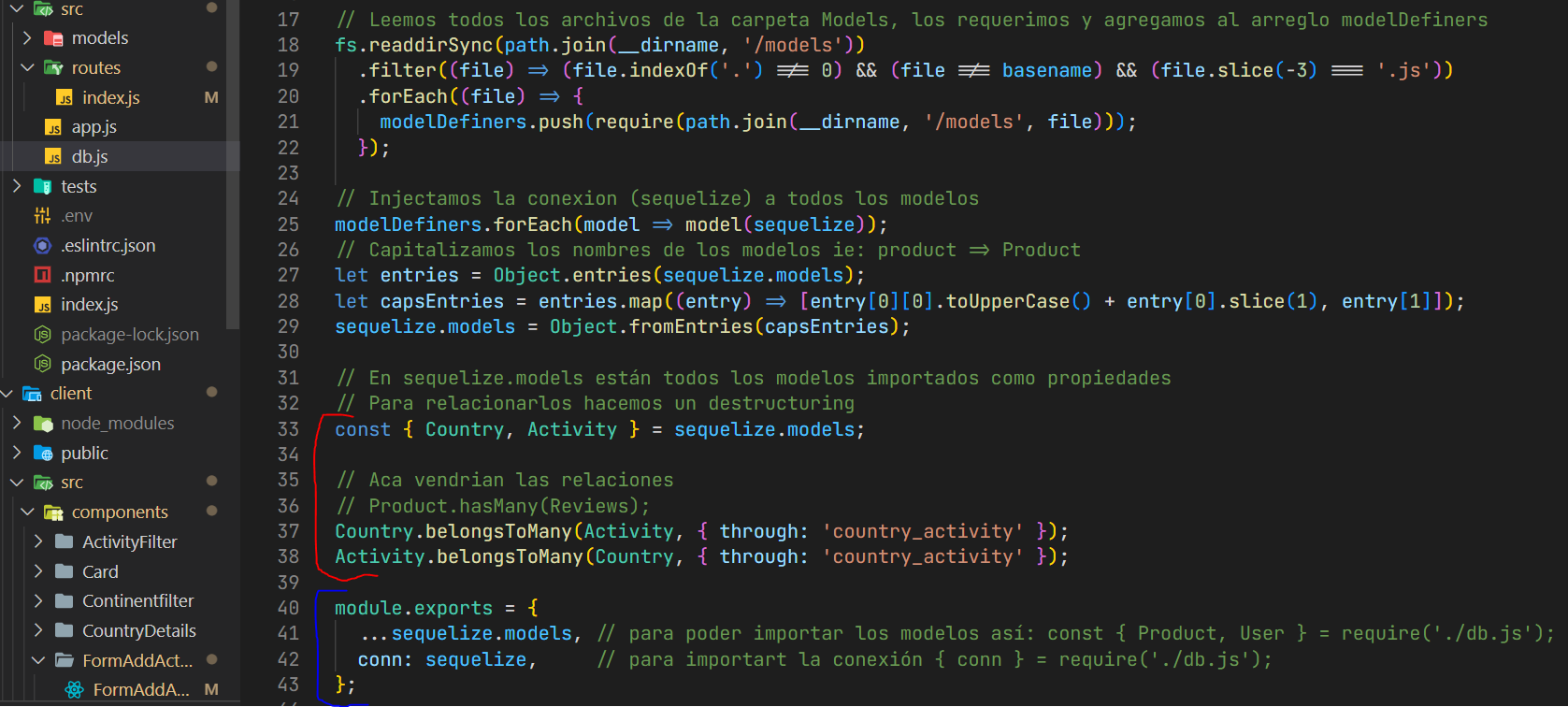
Usamos validate: {} para validar que solo reciba letras

DB

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Acá nos conectamos a la base de datos dando nuestros datos de conexión los cuales está en .env



El código arriba del paréntesis rojo se encarga de traer los archivos de los modelos, leerlos y conectarlos con Sequelize

Dentro de las llaves rojas tomamos los modelos importados y realizamos la asociación que en este caso es de muchos a muchos y creamos una tabla intermedia llamada country\_activity que tendrá los id del modelo Activity y Country.

Dentro de las llaves azules exportamos los modelos sequelize y la conexión de la base de datos.

Routes

Texto

Descripción generada automáticamente

Agregar la data del api de los países, de manera individual en cada una de las filas.

Captura de pantalla con la imagen de una pantalla

Descripción generada automáticamente

Traemos todos los países previamente cargados a la base de datos, y le decimos que de la información traida me ignore los campos de momento de agregado y actualizado a la vez que la tabla intermedia. A su vez me incluya la información en la tabla de actividades, está se van a comunicar vía su id de cada uno que las conecta. Y finalmente la función hace un return con dicha data.

Texto

Descripción generada automáticamente

En el primer end-point va a mostrar todos los países traídos de la BD y vamos a validad si viene algún valor por medio de la query, este será en caso de que el usuario desee buscar un país en específico y no toda la lista de países en la BD, a su vez tendrá un método lowerCase para que no importe si lo busca en minúscula o mayúscula.

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

Por medio de params recibimos el id de cada país, y este a su vez nos devuelve la data de dicho país en la BD, esto lo usaremos para ver los detalles de los países.

Texto

Descripción generada automáticamente

End point que nos trae las actividades de la BD de la tabla activities, donde solo tare el name y el id. Si no hay actividades trae un arreglo con un solo elemento que dice “No hay actividades”.

Texto

Descripción generada automáticamente

Recibimos unos valores por body(vendrán del formulario). Si el nombre de la actividad(se especifica en el where) no existe en la base de datos cree una fila con dicha información que asignara a su correspondiente columna las cuales son nombre, dificultad, duración y temporada.

Una vez creada la actividad le pido que vincule dicha actividad con una o múltiples ciudades especificadas por su id.

No olviddr sobre las funciones asíncronas(async await).

SERVER

Texto

Descripción generada automáticamente

Requerimos express, requerimos los middlewares necesarios y los aplicamos a express, a su vez que asignamos dominios para el acceso y evitar incidentes de CORS (Intercambio de Recursos de Origen Cruzado). Y al final mensajes de error en caso de tener problemas con la conexión.

INDEX API

Captura de pantalla de computadora

Descripción generada automáticamente

Requerimos el módulo de conexión a la base de datos(conn) y el servidor con express. force: true elimina los registros en la base de datos cada vez que se cargue el servidor. Va a correr el servidor en el puerto 3001 y devolverá un mensaje una vez conectado o que falle.