



Una prenda tiene una categoría, una categoría tiene muchas prendas.

Lo importante son las prendas, estas tendrán la clave foránea de las categorías. Y en estas usaremos la función belongsTo (pertenece).





Por un tema de categorías

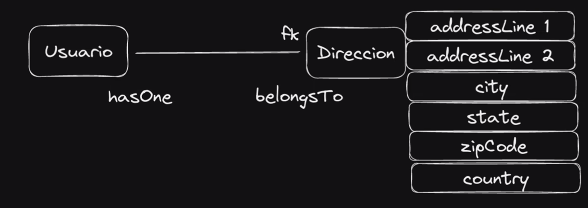
Funcions hasOne

1:N



No tengo manera de ver desde categorías que productos tengo, en este caso producto tendría mas “categoría”. En cambio…

1:1



En este caso la llave foránea va del lado de dirección para porque es la menos importante y asi no ensuciamos la tabla Usuarios. Entonces usamos hasOne y del otro lado belongsTo







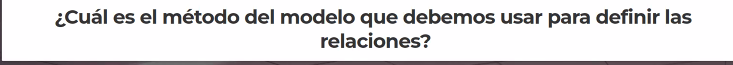


No pasa nada, solo que del lado del modelo que no hice la relación no podría pedir cosas.





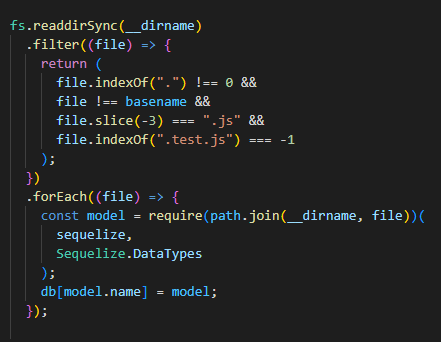
Delete() es el método de HTTPS en sequelize en comando es destroy()



Associate().

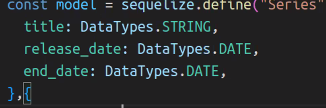
Explicación con el visual studio de la clase:

Como creo un modelo: exporto una función que resive sequelize y Data Types. Estos dos parámetros nos lo da index.js que a su vez solo lee los archivos que sean modelos, hace filtros para eso

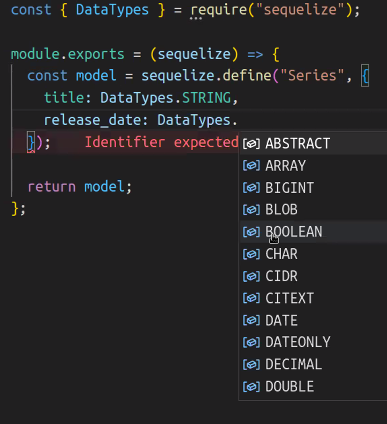


Creamos la varialbe del modelo que la vamsoa exportar al final.

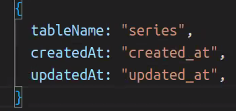
Primero le pasamos el nombre de la tabla



Segundo las columnas con sus configuraciones. Para que datatypes nos salte los autocompletados tengo que requerirlo por separado no dentro del module.export



Tercer parámetro la configuración: nombre de la tabla (), si tiene o no timestamps (en este caso si vamos a usar las columnas entonces aclaramos como se llaman)



Ahora tenemos una vista que vuestra todas las series. En el controlador requerimos el modelo



La lógica de buscar todas las series. Como un pedido a la DB tiene que ser asincrónico unamos una promesa .then()