

NEST 06

Crear servicio SEED para cargar datos

- Para hacer esto voy a comentar los brands y los cars
- Ahora si hago un Get o cualquier petición no sale nada. Voy a implementar un comando para cargar la data

```
nest g res seed --no-spec
```

- El no-spec es para que no cree el archivo de test
- Borro los dto y entities, también borro las referencias a los dtos y todos los servicios del seed.controller menos el Get, lo renombro a runSeed()
- En seed.service borro lo mismo, renombro el servicio a populateDB()
- Quedan así, seed.controller

```
import { Controller, Get, Post, Body, Patch, Param, Delete } from
 '@nestjs/common';
import { SeedService } from './seed.service';

@Controller('seed')
export class SeedController {
  constructor(private readonly seedService: SeedService) {}

  @Get()
  runSeed() {
    return this.seedService.populateDB();
  }
}
```

- seed.service

```
import { Injectable } from '@nestjs/common';

@Injectable()
export class SeedService {

  populateDB() {
    return "SEED executed";
  }
}
```

```
}
```

- Hago la prueba de que funcione con una petición Get

```
http://localhost:3000/sed
```

- cars y brands (las dbs ficticias que son propiedades de clase) son propiedades privadas. Debo encontrar la forma de exponerlas para cargar la data

Preparar servicios para insertar SEED

- Dentro de /seed creo una carpeta llamada data
- Dentro creo el archivo cars.seed.ts
- Creo varios items en CARS_SEED de tipo Car. Cómo es un arreglo, lo añado en tipado

```
import { v4 as uuid } from 'uuid'
import { Car } from "src/cars/interfaces/car.interface";

export const CARS_SEED: Car[] = [
  {
    id: uuid(),
    brand: "SEAT",
    model: "Ibiza",
  },
  {
    id: uuid(),
    brand: "Honda",
    model: "Civic",
  },
  {
    id: uuid(),
    brand: "Wolkskswagen",
    model: "Polo",
  }
]
```

- Creo otro archivo en /seed/data llamado brands.seed

```
import { v4 as uuid } from 'uuid'
import { Brand } from "src/brands/entities/brand.entity";

export const BRANDS_SEED: Brand[] = [
  {
    id: uuid(),
    name: "Volvo",
  }
]
```

```

        createdAt: new Date().getTime()
      },
      {
        id: uuid(),
        name: "Honda",
        createdAt: new Date().getTime()
      },
      {
        id: uuid(),
        name: "Toyota",
        createdAt: new Date().getTime()
      },
      {
        id: uuid(),
        name: "Suzuki",
        createdAt: new Date().getTime()
      }
    ]
  }
}

```

- El servicio populateDB en seed.service necesita trabajar con inyección de dependencias con los otros servicios
- Como son simples arreglos podría importarlos directamente en el servicio y retornarlo, pero lo que quiero retornar es un "Seed executed succesfull"
- Necesito poder crear un par de métodos para cargar la información
- En el cars.service creo un nuevo método llamado fillCarsWithSeedData()

```

fillCarsWithSeedData(cars: Car[]){
  this.cars = cars;
}

```

- Esto no es algo que vaya a hacer en la vida real porque lo insertaría directamente en la base de datos
- Hago lo mismo en el archivo brands.service
- Ahora tengo que llamar el brands.service y el cars.service desde el seed.service, para que después de ejecutar la semilla tenga la data

Inyectar servicios en otros servicios

- Voy a seed.service e importo el CarsService, creo una propiedad privada con este servicio
- Ahora solo tengo que llamar al método fillCarsWithSeedData con el CARS_SEED

```

import { Injectable } from '@nestjs/common';
import { CarsService } from 'src/cars/cars.service';
import { CARS_SEED } from './data/cars.seed';

```

```

@Injectable()
export class SeedService {

  constructor(

    private readonly carsService: CarsService,

  ){}

  populateDB() {
    this.carsService.fillCarsWithSeedData(CARS_SEED)

    return 'SEED EXECUTED';
  }
}

```

- Esto **da error**. CarsService es un provider porque tiene el decorador de @injectable, pero para poder importarlo debe de estar en el **mismo módulo**
- No está en el mismo módulo, por ello hay que exportarlo en el módulo e importarlo
- cars.module:

```

import { Module } from '@nestjs/common';
import { CarsController } from './cars.controller';
import { CarsService } from './cars.service';

@Module({
  controllers: [CarsController],
  providers: [CarsService],
  exports:[CarsService]
})
export class CarsModule {}

```

- seed.module:

```

import { Module } from '@nestjs/common';
import { SeedService } from './seed.service';
import { SeedController } from './seed.controller';
import { CarsModule } from 'src/cars/cars.module';

@Module({
  controllers: [SeedController],
  providers: [SeedService],
  imports: [CarsModule]
})
export class SeedModule {}

```

- Ahora ejecuto el Get del endpoint de seed, y luego el Get de cars y obtengo la data
- El mismo proceso con brands (^ ^)