Simple product api

Build a simple product api in TypeScript, which can manage product and producer entities using MongoDB.

Preferred a graphql based api, but if you are not familiar with graphql, a simple rest is perfect as well.

Do not use Apollo, NestJS, Prisma or any feature rich frameworks in your solution please.

Using `express`, `graphql`, `mongodb`, `mongoose` or other basic libraries (including csv parsers) are allowed of course.

Entities

The api manages the following two entities:

Product

- `_id: ObjectId!`
- `vintage: String!`
- `name: String!`
- `producerId: ObjectId!`
- `producer: Producer!`

Producer

- `_id: ObjectId!`
- `name: String!`
- `country: String`
- `region: String`

1st Task

Queries

- Query a single product by its `_id`.
- Query products by producer `_id`.

Only the single query (by `_id`) must be able to join the producer by the graphql query, eg:

```
query {
  product(_id: "...") {
    name
    producer {
      name
      country
      region
    }
  }
}
```

(If you work with REST instead of graphql, use a query param for this purpose.)

Mutations

- Create multiple products and return them
- Update a single product and return it
- Delete multiple products and return `true`

2nd Task

Create a mutation which

- returns `true` before it does anything (the http request should not await to finish the process),
- starts fetching this CSV file as a NodeJS stream,
- groups by `Vintage` + `Product Name` + `Producer` (as a unique field combination on products) and proceeds an upsert into the MongoDB. This grouping and upsert shall happen in batches (~100 document / request) within the stream.

If the stream aborts for some reason, it should not cause an unhandled rejection. Just log it to the console and be ready for accepting new connections.