

54. Add Mock Data to the Database

#mockdata #database #typeorm #migration #sql

Generate the mock data

- mockaroo.com
- With mockaroo, you can fill out the **columns of your database table** that you want to insert data into and it will generate fake data for you
- We click "download" to download the data as a **.sql** file
- **Note that** we wrap **createdAt** with quotes, because it contains an uppercase letter, and if we don't, it will be converted to lowercase
- **Note that** we use a date range for **createdAt** because we don't want all posts to have same **createdAt** value

The screenshot shows the mockaroo.com website interface. At the top, there's a green header with the mockaroo logo and navigation links like PRICING and SIGN IN. Below the header, there's a table configuration section with columns: Field Name, Type, and Options. The table has four rows: 'title' (Movie Title), 'text' (Paragraphs), '"creatorId"' (Custom List), and '"createdAt"' (Datetime). Each row has input fields for options like 'blank', 'at least', 'but no more than', 'random', and 'blank'. Below the table, there's a section for '# Rows: 100', 'Format: SQL', 'Table Name: post', and a checkbox 'include create table'. At the bottom, there's a 'Download Data' button, a 'Preview' button, and a 'More' dropdown menu.

Migrate the mock data into database

- Create a new **migration**. the **-n flag** allows us to give it a name

```
npx typeorm migration:create -n FakeData
```

- Create a **/src/migrations** folder and move the new **migration file** there
- Now copy the contents of the **.sql** file and paste it into this **migration file** as such:

/migrations/1718394969560-FakePosts.ts

```
import { MigrationInterface, QueryRunner } from "typeorm";
```

```
export class FakePosts1718394969560 implements MigrationInterface {  
  public async up(queryRunner: QueryRunner): Promise<void> {  
    // query created with mockaroo.com  
    await queryRunner.query(`PASTE THE SQL FILE CONTENTS HERE`); // mind the  
    back-tick instead of single quotes  
  }  
  
  public async down(_: /*queryRunner*/ QueryRunner): Promise<void> {}  
}
```

- And when we restart the server, the `await conn.runMigrations();` line in `index.tsx` will automatically run this `migration` and load this `mockdata` in to the `database`