ORMs part 3

Migrations, fixtures, and more

Sequelize CLI

Sequelize CLI

- `npm install --save sequelize-cli`
- “CLI” stands for **command-line interface**
- The **`npm`** program is a CLI, **`docker`** is a CLI too
- **`sequelize-cli`** will allow us to run some database-related tasks

Sequelize CLI

- `npx sequelize-cli init` to scaffold a new project

- `npx sequelize-cli model:generate --name User --attributes name:string,email:string` to create a **new model**
  - This also adds an initial migration for this model!
  - It places the model in its own file, in a special `models` folder

- `npx sequelize-cli db:migrate` to migrate the database structure

Migrations

Migrations

- Update the structure of your database, while it contains production data

- Each migration has an “up” and “down” half
  - The “down” half should undo whatever the “up” half does, if it is reversible!
  - This is so that you can quickly revert problematic changes or “go back” to a version
  - Proper reversible migrations are really hard/not ubiquitous in the real world!

- Sequelize provides functions for all sorts of **DDL** tasks like **`ALTER TABLE`**

- http://docs.sequelizejs.com/class/lib/query-interface.js~QueryInterface.html

Fixtures (Sequelize “seeders”)

Fixtures

- Database fixtures provide test data for development

- `npx sequelize-cli seed:generate --name demo-user`
- `npx sequelize-cli db:seed:all`
- `npx sequelize-cli db:seed --seed name-of-seed`
- `npx sequelize-cli db:seed:undo --seed name-of-seed`