



Atualizando o TypeORM no Rentx

Com a versão 0.3.0 algumas coisas deixaram de funcionar, outras mudaram seu funcionamento. Nesse guia rápido, vamos passar por alguns pontos essenciais que necessitam de atualização no TypeORM na aplicação Rentx, desenvolvida no Ignite Node.js.

⚠ Antes de tudo, é extremamente importante que leiam com muita atenção o changelog da versão 0.3.0, pois neste guia **não** será explicada todas as alterações da versão, apenas as necessárias para refatorar o Rentx.

Link: [Release 0.3.0 - typeorm/typeorm \(github.com\)](#)

Pontos de atenção

Conexão e configuração

O arquivo `ormconfig.` foi descontinuado. A nova forma de conexão e configuração é por meio da classe `DataSource`.

Essa classe pode receber as `entities` e `migrations`, assim como na configuração antiga, por meio do array de string (caminho para serem carregados), mas essa funcionalidade já está como depreciada e será removida nas versões futuras, por isso o ideal é informar diretamente cada classe no array.

Exemplo:

```
const dataSource = new DataSource({
  type: "postgres",
  port: 5432,
  username: "docker",
  password: "ignite",
  database: "rentx",
  entities: [
    Category,
    // ...
  ],
  migrations: [
    CreateCategories1616082124654,
    // ...
  ],
});
```

Para iniciar a conexão é preciso executar o método `initialize` da instância da classe.

Para encerrar a conexão não é mais o método `close`, e sim o `destroy`.

CLI

Todos os comandos da cli também não suportam mais o arquivo `ormconfig.`, agora é preciso informar o caminho para um arquivo que exporte por padrão uma instância da classe `DataSource`.

Exemplo:

```
yarn typeorm -d ./src/shared/infra/typeorm/index.ts migration:run
```

Repositórios

Para acessar os repositórios do TypeORM, antes era importado o método `getRepository`, essa forma está depreciada. Para recuperar os repositórios é preciso importar o `DataSource` e utilizar o método `getRepository` por ele.

Exemplo:

```
import dataSource from "@shared/infra/typeorm";
import { Category } from "../entities/Category";

const categoriesRepository = dataSource.getRepository(Category);
```

O método `findOne` não existe mais, então quando precisar buscar um único registro, deve utilizar o `findOneBy`, passando um objeto de configurações, com a condição a ser realizada.

Exemplo:

```
const user = await this.repository.findOneBy({ id });
```

O método `findByIds` não existe mais, então para buscar por um array de ids é preciso utilizar o operador `In`.

Exemplo:

```
import { In } from "typeorm";

const specifications = await this.repository.findBy({ id: In(ids) });
```

Concluindo


CONCLUSÃO

Essas são as principais alterações que se farão necessárias para atualizar o TypeORM. É claro que outras mudanças eventuais (tipos, importações, scripts) também são necessárias, mas decidimos não abordar nesse documento para não deixar extenso e complexo.

Para facilitar a atualização da sua aplicação, disponibilizamos abaixo o link direto para o commit efetuando as atualizações necessárias.




chore: update typeorm version - rocketseat-education/ignite-nodejs...


Aplicação de aluguel de carros desenvolvida no Ignite Node.js - chore: update typeorm version - rocketseat-education/ignite-nodejs-rentx@9467254

 <https://github.com/rocketseat-education/ignite-nodejs-rentx/commit/9467254>

chore: update typeorm version

Elton Mota - rocketseat-education/ignite-nodejs-rentx@9467254



Espero ter ajudado, abraços 💜