

MIGRAZIOAK

Zer dira migrazioak? Datu basearen bertsio kontrola eramateko erabiltzen dira. Modu honetan zeozer aldatuz gero, aldaketak gordeta geratuko dira beste norbaitek erabili ahal izateko edo arazoren baten bat egotekotan, aurreko puntura bueltatzea errezago izango da.

Migrazioak datu basean ditugun taulak kudeatzeko ere erabili ditzakegu, hau da, taula berriak sortzeko, taularen eremuren bat aldatu ahal izateko edota ezabatu ahal izateko.

Migrazioak `database>migrations` karpetan gordeko dira. Migrazio bakoitzak php baten egongo da, eta horretan ikusi dezakegunez Migration klase bat dago 2 metodo dituenak: `up()` eta `down()`. Migrazioa exekutatzekoan `up()` metodoari deituko dio, eta reset edo refresh edo horrelakoetan, `down()` metodoari deituko zaio.

Migrazioekin lan egiteko, artisaiek baditu hainbat metodo: `make:migration`, `migrate:refresh`, `migrate:reset` eta `migrate:rollback`

make:migration

Metodo honekin migrazio berri bat sortuko dugu

migrate:rollback

Kasu honetan, reset bat egingo du, baina bakarrik azkenik exekutatu den loteari. Migrazioak zein lotetan exekutatu den ikusteko, gure datu basera joan behar dugu eta migrations taulan `batch` eremua begiratu. Rollback-ek ere `down()` metodoa exekutatu du.

migrate:reset

Metodo honek taula guztiak ezabatuko ditu, migrazio taula izan ezik, hau da, migrazioak duen `down()` metodoa exekutatu du

migrate:refresh

Metodo honek lehenago reset bat egingo du, hau da `down()` metodoari deituko dio eta gero berriro exekutatu ditu denak, hau da `up()` metodoari deituko dio.

Bai reset, bai refresh metodoek gure taulak ezabatuko ditu, hau da, barruan dituen datu guztiak ere ezabatu egingo ditu, beraz kontuz ibili behar dugu hauek exekutatzeko orduan. Eta kontutan izan ere, ditugun migrazio guztiak exekutatu direla.

migrate:fresh

Kasu honetan, lehenengo taula guztiak ezabatuko ditu, ez du `down()` metodoari deituko eta gero, `up()` metodoari deituko dio.

Hau adibidez beharrezkoa izango da taula baten eremuak aldatu direnean, baina modu hau ez da oso gomendagarria. Metodo hau bakarrik da gomendagarria proiektua garapen momentuan dagoenean, eta ez produkzioan dagoenean jada.

****** Praktikan refresh eta fresh ia berdinak dira, baina refresh kasuan zeozer txarto idazteketan `down()` metodoan ez luke taula ezabatuko, aldiz bestearekin, beti ezabatuko ditu.

Zelan egin taula baten eremu aldaketa taulak datuak izanda?

Taula baten eremuak aldatzeko, migrazioa sortuko genuke ere.

Adibidez, taula baten eremu berri bat sartzeko, migrazio berri bat sortu beharko genuke. Lehen esan dugun moduan, migrazioaren izena edozein izan daiteke, baina "konbentzioz" horrela idatziko genuke:

```
php artisan make:migration add_avatar_to_users_table
```

```
public function up(): void
{
    Schema::table('users', function (Blueprint $table) {
        //
    });
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::table('users', function (Blueprint $table) {
        //
    });
}
```

Sortuko zaigu migrazio dokumentu bat aurreko metodoekin, beraz hemen eremua gehituko dugu create kasuan moduan. Taulan datuak egongo direnez, eremua sortzerakoan jarri behar dugu null izan daitekeela, akatsarik ez emateko. Hau `up()` metodoan idatziko dugu.

```
$table->string('avatar')->nullable();
```

`down()` metodoan ere zutabea ezabatu beharko dugu

```
$table->dropColumn('avatar');
```

Migrazio hau exekutatzekoan, avatar zutabea gehitu du, baina taularen bukaeran. Horrela nahi ez izatekotan, rollback egingo dugu eta gero aldaketa migrazioan

```
php artisan migrate
```

Demagun email eremuaren ostean jarri nahi dugula:

```
$table->string('avatar')->nullable()->after('email');
```

Dokumentazio ofiziala hurrengo esteka honetan duzue: <https://laravel.com/docs/10.x/migrations>

Hurrengoa: Eloquent