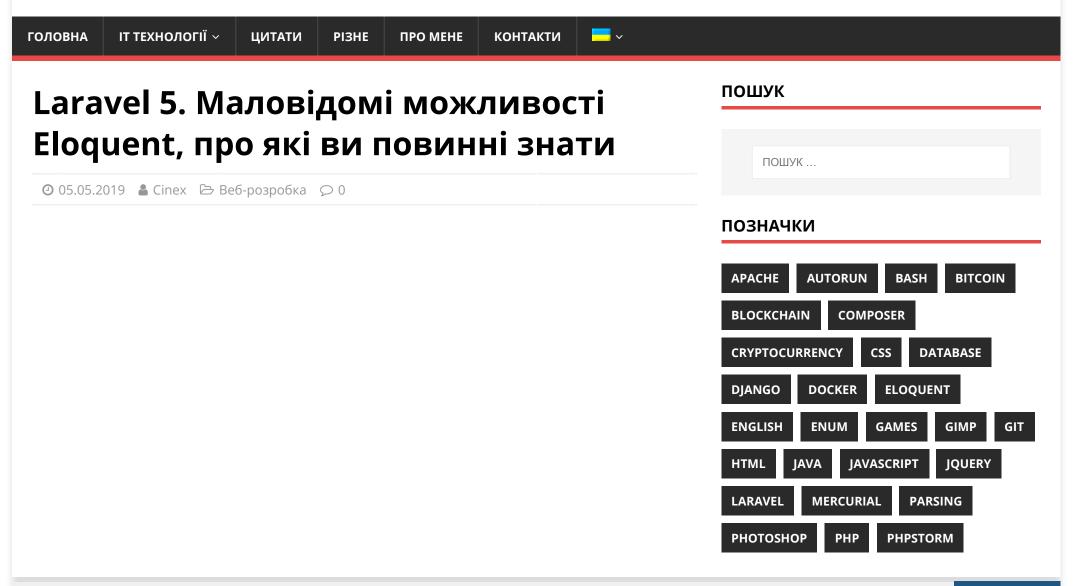
THE CINEX

ВСЕ ДЛЯ ВСІХ



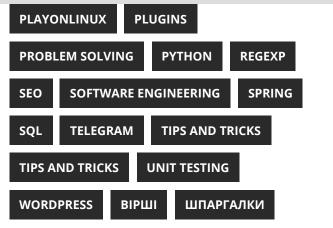


Laravel настільки багатий на можливості, що багато з них навіть не описані в офіційній документації. У цій статті ви дізнаєтеся про деякі з них.

1. Отримати початкове значення атрибута

Після зміни атрибута в eloquent-об'єкті ви можете отримати початкове значення за допомогою методу getOriginal()

```
1 $user = App\User::first();
2 $user->name; //John
3
4 $user->name = "Peter"; //Peter
5
6 $user->getOriginal('name'); //John
```



останні нотатки

20 Laravel Eloquent порад і трюків 15.12.2019

Як зробити скріншот сайту по URL на PHP 04.08.2019

Docker Cheat Sheet

05.07.2019

Гарячі клавіші Ubuntu Linux

01.07.2019

Git happens! 6 типових помилок Git і як їх виправити

22.06.2019

НЕДАВНІ КОМЕНТАРІ

```
7 $user->getOriginal(); //Original $user record
```

2. Визначити, чи змінена модель

Можна за допомогою методу isDirty()

```
1 $user = App\User::first();
2 $user->isDirty(); //false
3
4 $user->name = "Peter";
5 $user->isDirty(); //true
```

Також можна перевірити зміна конкретного атрибута

```
1 $user->isDirty('name'); //true
2 $user->isDirty('age'); //false
```

3. Отримати змінені атрибути

Можна за допомогою методу getChanges().

4. Зміна назви стовпчика deleted_at

За замовчуванням при м'якому видаленні (soft delete) Laravel працює зі стовпцем deleted_at. Ви можете змінити назву, визначивши властивість DELETED_AT.

```
1 class User extends Model
2 {
```

```
3  use SoftDeletes;
4
5  * The name of the "deleted at" column.
6  *
7  * @var string
8  */
9  const DELETED_AT = 'is_deleted';
10 }
```

Другий спосіб:

```
class User extends Model
{
    use SoftDeletes;
    public function getDeletedAtColumn()
    {
        return 'is_deleted';
    }
}
```

5. Збереження моделі разом зі зв'язками

Ви можете зберегти модель з відповідними зв'язками за допомогою методу push()

```
■ <> 
□ □ PHP
1 class User extends Model
2 {
     public function phone()
4
       return $this->hasOne('App\Phone');
5
6
7
8
   $user = User::first();
10 $user->name = "Peter";
11
12 $user->phone->number = '1234567890';
13
14 $user->push(); // оновлення і user, і phone в базі
```

6. Завантаження оновленої моделі

Якщо модель змінилася в базі даних, то довантажити зміни можна за допомогою методу fresh()

```
1 $user = App\User::first();
2 $user->name; // John
3
4 // У цей момент в іншому потоці User змінився.
5 // Наприклад, 'name' -> Peter.
6
7 $updatedUser = $user->fresh();
8 $updatedUser->name; // Peter
9
10 $user->name; // John
```

7. Оновлення змінити модель

Оновити об'єкт з бази даних можна за допомогою методу refresh(). Також буде оновлено всі зв'язки.

```
1 $user = App\User::first();
2 $user->name; // John
3
4 // у цей момент в іншому потоці user змінився.
5 // Наприклад, 'name' -> Peter.
6
7 $user->refresh();
8 $user->name; // Peter
```

8. Визначити, чи індентичні моделі

Визначити, чи однакові id у моделей i чи належать вони одній таблиці можна за допомогою методу is()

```
1 $user = App\User::find(1);
```

```
2 $sameUser = App\User::find(1);
3 $diffUser = App\User::find(2);
4
5 $user->is($sameUser); // true
6 $user->is($diffUser); // false
```

9. Клонування моделі

Метод replicate() створює копію моделі як нову сутність, яку потрібно потім зберегти.

```
1 $user = App\User::find(1);
2 $newUser = $user->replicate();
3
4 $newUser->save();
```

10. Вказівка атрибутів в методі find()

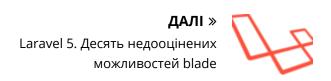
У методах find() i findOrFail() другим аргументом можна вказати атрибути для вибірки з бази

```
1 $user = App\User::find(1, ['name', 'age']);
2 $user = App\User::findOrFail(1, ['name', 'age']);
```

Джерело







ЗАЛИШТЕ ПЕРШИЙ КОМЕНТАР

Залишити коментар

Вашу адресу електронної пошти не буде опубліковано		
Коментувати		
	//	
lм'я*		
Email *		
🔲 Збережіть моє ім'я, електронну пошту у цьому веб-переглядачі наступного разу, коли я		
коментуватиму.		
ОПУБЛІКУВАТИ КОМЕНТАР		

Copyright © 2020