



10 mažiau žinomų Eloquent galimybių

1. Mutators

```
class User extends Model {  
    /**  
     * Get the user's first name.  
     *  
     * @param string $value  
     * @return string  
     */  
    public function getFirstNameAttribute($value) {  
        return ucfirst($value);  
    }  
}
```

Mutators

```
class User extends Model {  
    /**  
     * Set the user's first name.  
     *  
     * @param string $value  
     * @return string  
     */  
    public function setFirstNameAttribute($value) {  
        $this->attributes['first_name'] = strtolower($value);  
    }  
}
```

2. Casters

```
class User extends Model {  
    /**  
     * The attributes that should be casted to native types.  
     *  
     * @var array  
     */  
    protected $casts = [  
        'is_admin' => 'boolean',  
    ];  
}
```

Casters

- integer
- real
- float
- double
- string
- boolean
- object
- array
- collection
- date
- datetime

3. Eager loading

Authors:

Id	Name
1	Modestas

Books:

Id	Author_id	Title
1	1	Laravel
2	1	Tips
3	2	Tricks

Eager loading

```
class Book extends Model {  
    /**  
     * Get the author that wrote the book.  
     */  
    public function author() {  
        return $this->belongsTo('App\Author');  
    }  
}
```

Eager loading

```
$books = App\Book::all();
```

```
foreach ($books as $book) {  
    echo $book->author->name;  
}
```


Eager loading

101 statements were executed, 10 of which were duplicated

```
select * from `books`
```

```
select * from `authors` where `authors`.`id` = '2' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '8' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '1' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '9' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '2' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '1' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '5' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '4' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '4' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '9' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '5' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '1' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '1' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '5' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '3' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '2' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '2' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '8' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '4' and `authors`.`id` is not null limit 1
```

```
select * from `authors` where `authors`.`id` = '4' and `authors`.`id` is not null limit 1
```

Eager loading

```
$books = App\Book::with('author')->get();
```

```
foreach ($books as $book) {  
    echo $book->author->name;  
}
```

Eager loading

2 statements were executed

```
select * from `books`
```

```
select * from `authors` where `authors`.`id` in ('2', '8', '1', '9', '5', '4', '3', '10', '6', '7')
```

Eager loading

```
$books = App\Book::with('author', 'publisher')->get();
```

Eager loading

```
$books = App\Book::with('author.contacts')->get();
```

Eager loading

```
$users = App\User::with([  
    'posts' => function ($query) {  
        $query->where('title', 'like', '%first%');  
    }  
])->get();
```

4. Eager loading ordering

```
public function author()  
{  
    return $this->hasOne('App\Authors')->orderBy('name');  
}
```

Eager loading ordering

```
$books = Books::with([  
    'author' => function ($query) {  
        $query->orderBy('name');  
    }  
])->get();
```


5. Chunk

```
$collection = collect([1, 2, 3, 4, 5, 6, 7]);
```

```
$chunks = $collection->chunk(4);
```

```
$chunks->toArray();
```

```
// [[1, 2, 3, 4], [5, 6, 7]]
```

Chunk

```
@foreach ($products->chunk(3) as $chunk)
    <div class="row">
        @foreach ($chunk as $product)
            <div class="col-xs-4">{{ $product->name }}</div>
        @endforeach
    </div>
@endforeach
```

6. Every

```
$collection = collect(['a', 'b', 'c', 'd', 'e', 'f']);
```

```
$collection->every(4);
```

```
// ['a', 'e']
```

```
$collection->every(4, 1);
```

```
// ['b', 'f']
```

7. Filter

```
$collection = collect([1, 2, 3, 4]);
```

```
$filtered = $collection->filter(function ($item) {  
    return $item > 2;  
});
```

```
$filtered->all();
```

```
// [3, 4]
```

8. KeyBy

```
$collection = collect([
    ['product_id' => 'prod-100', 'name' => 'desk'],
    ['product_id' => 'prod-200', 'name' => 'chair'],
]);
$keyed = $collection->keyBy('product_id');
$keyed->all();
/*
[
    'prod-100' => ['product_id' => 'prod-100', 'name' => 'Desk'],
    'prod-200' => ['product_id' => 'prod-200', 'name' => 'Chair'],
]
*/
```

9. Random

```
$collection = collect([1, 2, 3, 4, 5]);  
$collection->random();  
// 4 - (retrieved randomly)
```

```
$random = $collection->random(3);  
$random->all();  
// [2, 4, 5] - (retrieved randomly)
```

10. Where

```
$books = Books::where('title', $title)->first();
```

```
$books = Books::whereIn('id', [1, 2, 3])->get();
```

```
$books = Books::whereTitle($title)->first();
```

```
$books = Books::whereDay('date', '=', date("d"))->get();
```

```
$books = Books::whereMonth('date', '=', date("M"))->get();
```

```
$books = Books::whereYear('date', '=', date("Y"))->get();
```

Turite klausimų?