

Custom Query Builders

First, let's discuss what a query builder is. When you write something like that:

```
$query = Product::where(
    'name',
    'Domain-Driven Design with Laravel'
);
```

You're interacting with the `\Illuminate\Database\Eloquent\Builder` class. It contains your favorite Eloquent methods such as `where`, `firstWhere`, `latest`, etc.

But in Laravel, we don't write code like this:

```
Builder::where('description', 'LIKE', "%{$searchTerm}%");
```

We call the `where` method on the `Model` classes. So there has to be some connection between `Model` and `Builder`. This connection is the `newEloquentBuilder` in the base `Model` class:

```
/**
 * Create a new Eloquent query builder for the model.
 *
 * @param \Illuminate\Database\Query\Builder $query
 * @return \Illuminate\Database\Eloquent\Builder|static
 */
public function newEloquentBuilder($query)
{
    return new Builder($query);
}
```