

It returns a new `Builder` instance, and every time you interact with your model, this instance will be used. So when you call `Product::where` the `newEloquentBuilder` method will be called, it returns a new `Builder`.

We can extend this base `Builder` class and can create a custom query builder:

```
class MailBuilder extends Builder
{
    public function whereOpened(): self
    {
        return $this->whereNotNull('opened_at');
    }
}
```

This is a custom builder for the `Mail` model. The `whereOpened` method can be used as a scope. In fact, model scopes are just syntactic sugar. So this method can be used like this:

```
Mail::whereOpened()->get();
```

In the builder we need to return `self` to be able to chain Eloquent methods:

```
Mail::whereOpened()->where('title', 'First Mail')->get();
```

However, you don't have to write scope-like methods. In a query builder, you can do (almost) anything: