

```

        →get();
    }
}

```

It's a class for your database queries. That's it. So instead of writing your queries inside your controllers or models, you move them into a repository. Usually, each model has a repository, but that's not necessarily true every time.

You can use it like this:

```

class ProductSearchController extends Controller
{
    public function index(
        Request $request,
        ProductRepository $products
    ) {
        return $products→search(
            PriceRange::from($request),
            $request→search_term
        );
    }
}

```

So it seems like a good concept; why are they being hated? There are several problems:

- Usually, each (or the vast majority of) model has a repository, such as the `ProductRepository` in this example. So you're literally just moving a query from one place to another, from the model to the repository. After six months of development, the `ProductRepository` will become a 5000 lines monster. Just like the `Product` model would have become a 5000 lines monster.
- By definition, when using repository classes, **every** database query should end up in one of them. So you need to move single-use queries as well. You'll quickly end up with