

```

public function tags(): BelongsToMany
{
    return $this->belongsToMany(Tag::class);
}

public function form(): BelongsTo
{
    return $this->belongsTo(Form::class)
        ->withDefault();
}
}

```

At this point, there's nothing too interesting. A many-to-many tag and a one-to-many form relationship. In the form relationship, I use the `withDefault` method. By default, if a subscriber does not have a form (it's a nullable column), Laravel will return `null`, and it can cause some bugs and unnecessary `if` statements. Using the `withDefault` Laravel will return an empty `Form` object instead of `null`. It can be helpful in some situations.

Now, let's see the subscriber DTO. In code, I don't use the word 'DTO'. I call the namespace `DataTransferObjects`, and I suffix the classes with the term `Data`.

This is the `SubscriberData` class:

```

namespace Domain\Subscriber\DataTransferObjects;

use Spatie\LaravelData\Data;

class SubscriberData extends Data
{
    public function __construct(

```