

```

public static function fromModel(Subscriber $subscriber): self
{
    return self::from([
        ... $subscriber->toArray(),
        'tags' => Lazy::whenLoaded(
            'tags',
            $subscriber,
            fn () => TagData::collection($subscriber->tags)
        ),
        'form' => Lazy::whenLoaded(
            'form',
            $subscriber,
            fn () => FormData::from($subscriber->form)
        ),
    ]);
}

```

What is this `Lazy` thing? It's basically the same as the `whenLoaded` in HTTP resources. For example, the `$form` property will only be evaluated if the `form` relationship is loaded in the given `$subscriber` instance. Only then the callback of the `whenLoaded` will be executed. This helps us to avoid N+1 queries. Otherwise, when querying the subscribers and mapping each one to a data object, it would run two extra queries (loading the form and the tags) **for every subscriber**. This is why the `Lazy::whenLoaded` is such a valuable helper. To use this, we need to make one more change: