

I will talk about the action later, but now let's focus on the DTO. As you can see, we can inject the `SubscriberData` class into the method, and the package will automatically create a new instance from the request. But I also inject the `Request` itself to get the currently logged-in user.

I could put a user property in the `SubscriberData` but remember we will also use this class to return as a response, and I don't want to include sensitive user data in the responses if it's not necessary. And in most cases, the user is not needed on the frontend. Remember, when users log in to the application, they will only see their information. So there's no need to display to creator user's name or anything like that. For that reason, I left out the user property from the DTO.

By default, `laravel-data` will do a one-to-one mapping from the request to the data object. For each property within the data object, a value with the same key will be searched within the request values. This default behavior works in most cases. But if you take a look at the data class, it has properties like `tags` and `form`; meanwhile, in request, we expect keys like `tag_ids` and `form_id`. So how can we override the default mapping?

We need to define a `fromRequest` method in the data class:

```
public static function fromRequest(Request $request): self
{
    return self::from([
        ... $request->all(),
        'tags' => TagData::collection(
            Tag::whereIn('id', $request->collect('tag_ids'))->get()
        ),
        'form' => FormData::from(
            Form::findOrCreate($request->form_id)
        ),
    ]);
}
```