

This is a tiny application, but you can see the code that handles Departments is spread across **six different places**. It's grouped by technical attributes:

- Controllers
- Models
- Requests
- Resources
- Actions
- DTOs

Imagine a project with 200 models, 300 actions, 500 controllers, and 500+ routes. After a while, it becomes a nightmare. If you need to introduce a new attribute to the department model and accept it in your API, you probably have to change **five files in six different directories**.

By the way, have you ever heard a stakeholder saying something like: “Please work a little bit on the DepartmentController and add a new attribute to the DepartmentResource”? Of course not. They say things like: “On the department page, I need to see the number of employees within the department.” And from that information, you know that you probably need to tweak the DepartmentController and the DepartmentResource.

A developer is someone who translates the business language into code.

So, in my opinion, it is a good idea if the code is similar to the business language and follows the same structure. By using domains, we can achieve a much better structure: