

# Challenge 4

By: Mosh Hamedani

## Refactoring Towards the Clean Architecture

Our API is ready. However, there are two issues with the current implementation.

**1- API Contract:** VehicleResource works perfectly fine when creating/updating vehicles. But we can't use it when rendering a list of vehicles. There, we need the actual "make" and "model" of a vehicle, not just their IDs. With the current implementation, in order to render a list of vehicles, we need to make a few separate API calls, get the vehicles, makes and models and join them on the client. This is tedious and inefficient. Revisit the contract of the vehicles API to solve this issue.

**2- Coupling to Entity Framework:** Our current implementation is tightly coupled to Entity Framework. Use the repository and unit of work patterns to decouple the application from Entity Framework. You can learn more about these patterns and their implementation in the last section of my Entity Framework course:

<https://www.udemy.com/entity-framework-tutorial/>

## Dependency Injection

Use the **ConfigureServices()** method in the **Startup** class to register your interfaces and their implementations.

```
services.AddScoped<IRepository, Repository>();
```