1. A. Duplicated Code and Logic Code,

Making a same logic code in multiple method, where put same logic on add and edit to view. Extract the same code into separate method which help us to reuse that code.

B. Depedencies Injection

as far as i know, a technique for managing the way in which an object is formed when there are other objects that need it (Reusable). There are 2 term of Depedencies Injection:

* Service, this is where business handled logic, like validation, insert input from handler(which already converted in formatter) to struct model field. ex: input.name = user.name,send data to client(repository) get return from repository
* Client (repository), this place related to database, where Create, Read, Update, Delete handled, in this phase we get input from service,and execute them in which process we want (CRUD) and return result of executing back to service

Important, because the code and flow will be easy to maintenance, in service we handle business logic, and in client we handle CRUD And can reusable client and service

1. Post :

* v1 := router.Group("/api/v1")

{

Do

v1.GET("/campaigns/all"

v1.GET("/campaigns/all/:id"

Don’t

v1.POST("/campaigns/GetOrders"

v1.POST("/campaigns/GetOrder?id=:id”

}

Get :

* v1 := router.Group("/api/v1")

{

Do

//after deleted the data, i would find detail id that already deleted

v1.GET("/campaigns/:id "

//response : 410 Gone

Don’t

//after deleted the data, i would find detail id that already deleted

v1.GET("/campaigns/:id "

//response : 404 not found

}