Software Code Base Update: CI3 to CI4/Laravel

By MsM Robin

Codeigniter 3 is lighter framework with some limited features. It exactly not supports the MVC architecture but HMVC (Hierarchical Model-View-Controller). Whereas Codeigniter4 is also lighter and modern framework than CI3 with more features specially managing packages/tools thorough composer. Also, CI4 is completely re-write of the CI3 framework, supports full features of MVC and mostly inspired by Laravel architecture.

As a result, if we want to update our CI3 modules into CI4/Laravel there is some particular sections we need to keep in mind that:

# 1/ Configurations

This is would be the major concern if we eager to update CI3 modules into CI4/Laravel. Just because of all these 3 frameworks follows different architecture compare to each other.



## Base Controller

* **CI3**: Uses a basic structure without strict conventions.
* **CI4/Laravel**: Enforces a more structured approach with a base controller class. For instance, in CI4, you might extend BaseController, whereas in Laravel, you typically extend Controller.

## Rest Controller

* **CI3**: Lacks built-in support for RESTful controllers, requiring custom implementation.
* **CI4**: Provides a RESTful controller out of the box with features like resource routing.
* **Laravel**: Offers extensive support for RESTful APIs, including resource controllers and routes.

## Database configurations

* **CI3**: Database configurations are straightforward but limited.
* **CI4**: Introduces a more flexible and powerful database configuration system.
* **Laravel**: Offers robust database configuration options, including support for multiple connections and databases.

## 1.4 Filters

* **CI3**: Limited to hooks for pre- and post-processing.
* **CI4**: Implements a filter system that allows for pre- and post-processing of requests.
* **Laravel**: Utilizes middleware for handling request filtering, offering a more granular and powerful approach.

## 1.5 Session and Token managements

* **CI3**: Basic session handling with limited token management.
* **CI4**: Improved session management with better security features.
* **Laravel**: Comprehensive session and token management, including CSRF protection and API tokens.

## 1.6 Usages of Constants

* **CI3**: Constants are often defined in configuration files or manually within code.
* **CI4/Laravel**: Encourages the use of environment files (.env) for defining constants and configuration settings, promoting better security and flexibility.

# 2/ Routes

Routing differs significantly between CI3, CI4, and Laravel:

* **CI3**: Primarily uses auto-routing, which can be less secure and less flexible.
* **CI4**: Introduces a more structured routing system with support for both manual and auto-routing.
* **Laravel**: Offers a highly flexible and powerful routing system, including named routes, route groups, and middleware.

# 3/ Magic method to Constructors

In CI3 Controllers and Models are load/call/invoke by its magic method called load. But in CI4/Laravel can be called.

In CodeIgniter 3, it's not straightforward to call a method of one controller from another controller directly due to the way the framework is designed. However, there are a few ways you can achieve similar functionality:

1. **Libraries**: Common code can be moved to libraries.
2. **Helpers**: Common functions can be moved to helpers.
3. **Models**: Business logic can be moved to models.
4. **Modular Separation**: Use HMVC (Hierarchical Model-View-Controller) to enable modular separation.

Whereas CI4/Laravel follow same approach to create class instance directly on the construction or a method.

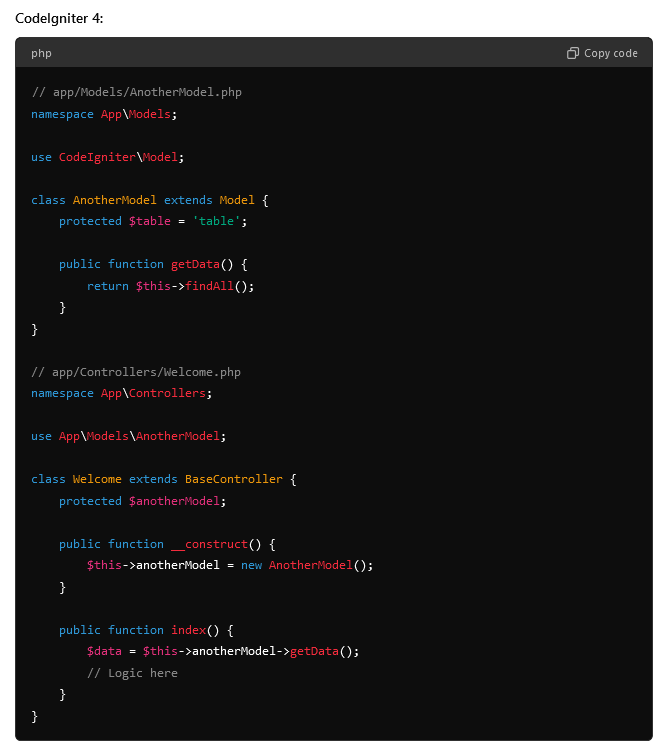


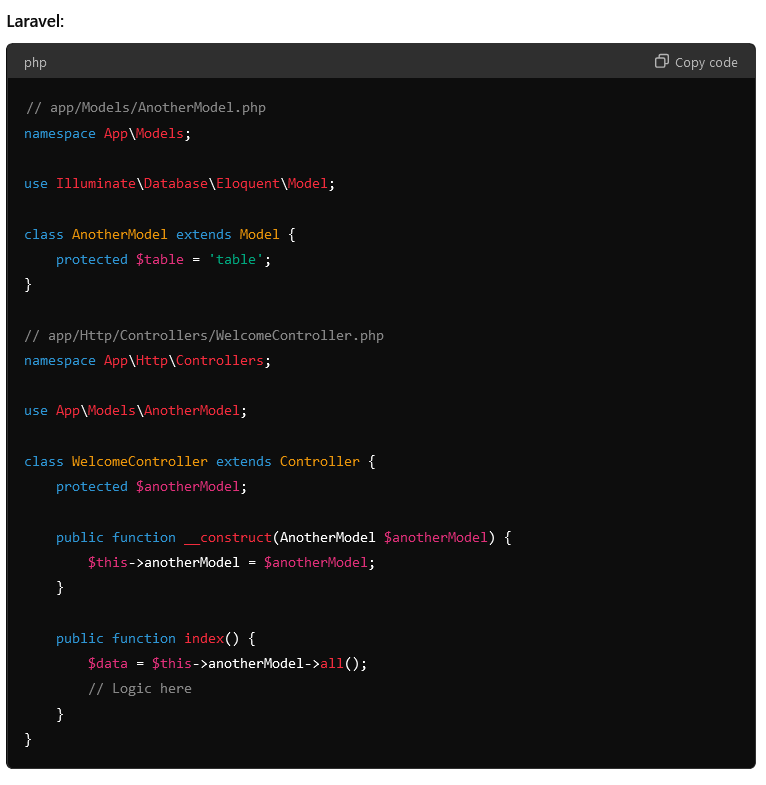


## 3.1 Calling/Invoking Another Controller to another.

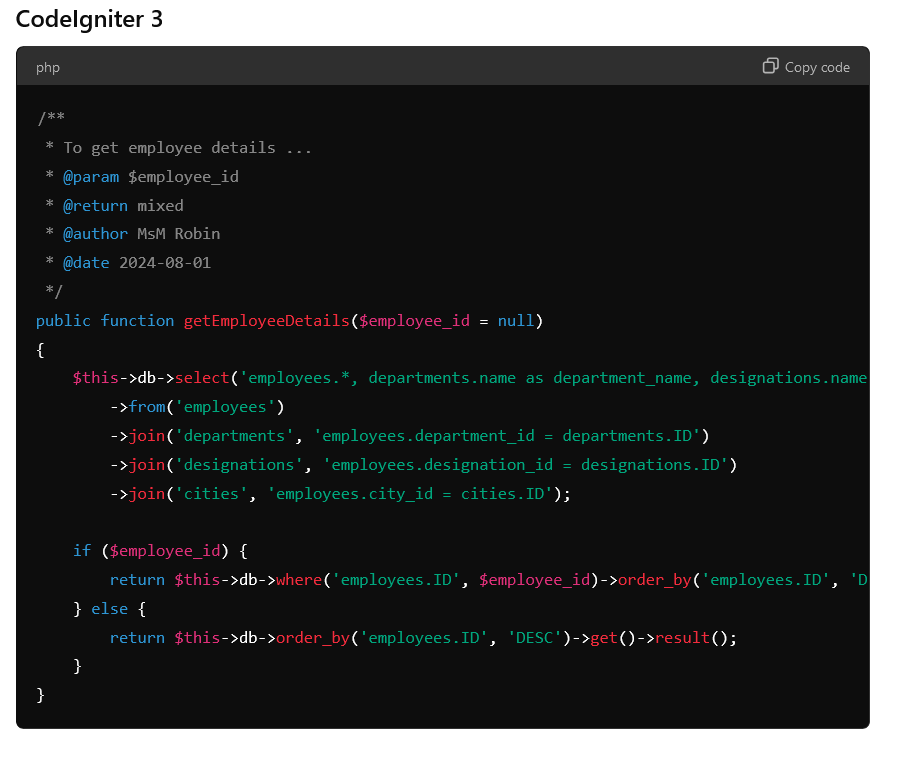
## 3.2 Calling/Invoking Another model to controller/model.



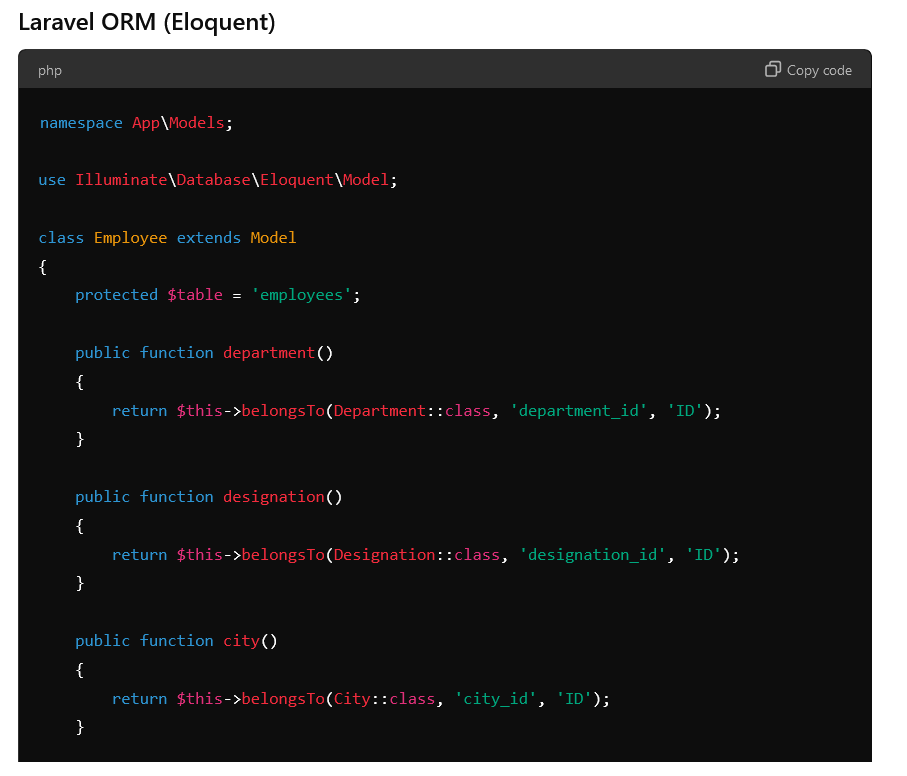




# 4/ Query builder methods











# 5/ Helpers, libraries to services