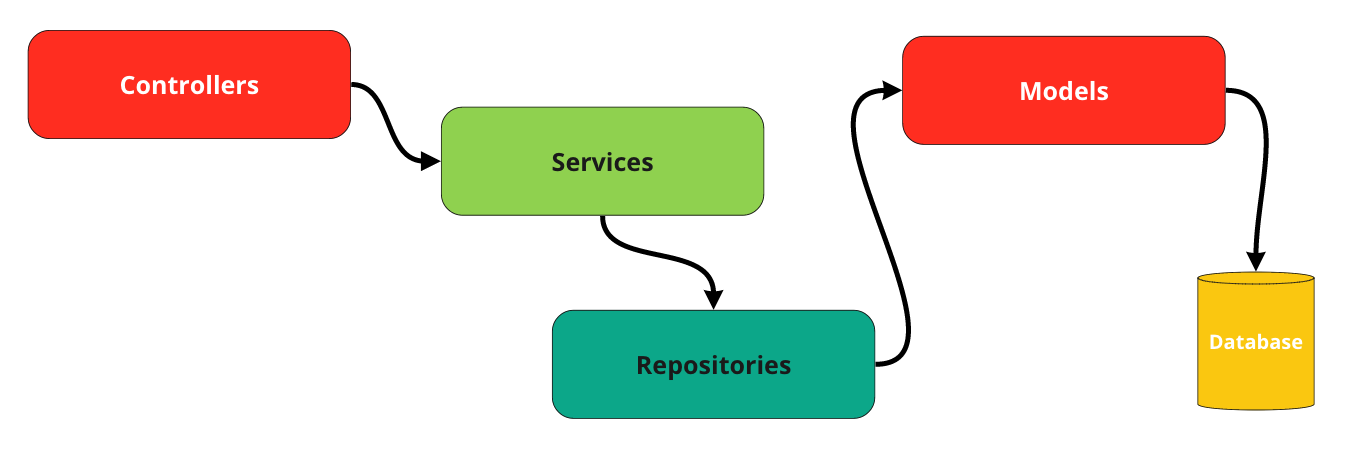
Service-Repository Design Pattern

**What is service repository pattern?**

Service and Repository is the break of model portion. The main idea to use **Service-Repository** Pattern in a Laravel application is **to create a bridge between models and controllers.** A **Repository** is data binding or query (CRUD) to supply access to data or objects. A **Service** is the business logic code or calculation. The **Controller** gives instruction to the **Service** and the service talks to the **Repository** to do some CRUD to the data from database.



**What is MVC?**

The **Model-View-Controller (MVC)** is an architectural pattern that separates an application into three main logical components: the **model**, the **view**, and the **controller**.

* Controller

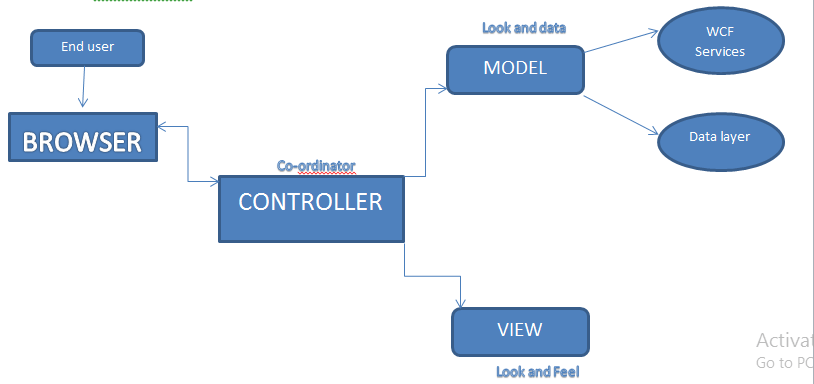
Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output (request and response).

* View

The View component is used for all the UI logic of the application. For example, text boxes, dropdowns, etc.

* Model

Model do data binding (query) and business logic (calculation). For example, a customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.



**Advantages of service repository pattern?**

* Business logic can be tested without the need for an external source
* Database access logic can be tested separately
* No duplication of code
* Centralizing the data access logic, so code maintainability is easier
* If you want to modify the data access logic or business access logic, you don't need to change the repository logic.

**Disadvantages of service repository pattern?**

* evolution of data is difficult and expensive.