



LARAVEL

LARAVEL NEDİR?

Laravel, MVC yapısında web uygulamaları geliştirme için tasarlanmış ücretsiz, açık kaynak PHP web uygulama iskeletidir.

LARAVELİN ÖZELLİKLERİ



Laravel, Paket Yönetim Sistemi(Dependency Manager) olarak Composer kullanır.



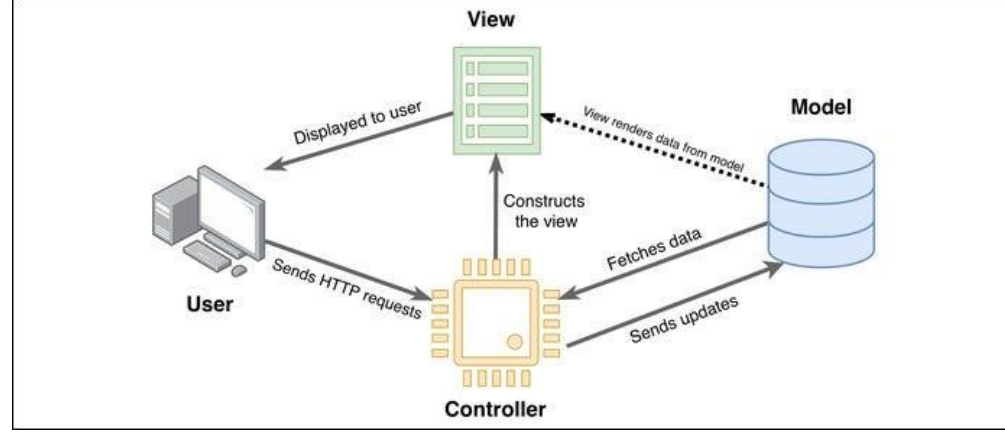
Laravel'in içinde ORM vardır. ORM yani Object Relational Mapping, veritabanınızdaki her tablonun karşısında bir modelin oluşturulmasıdır.



Yapısı gereği gerçekten çok sade ve temiz kod yazarak istediğiniz uygulamaları geliştirme imkanı verir.

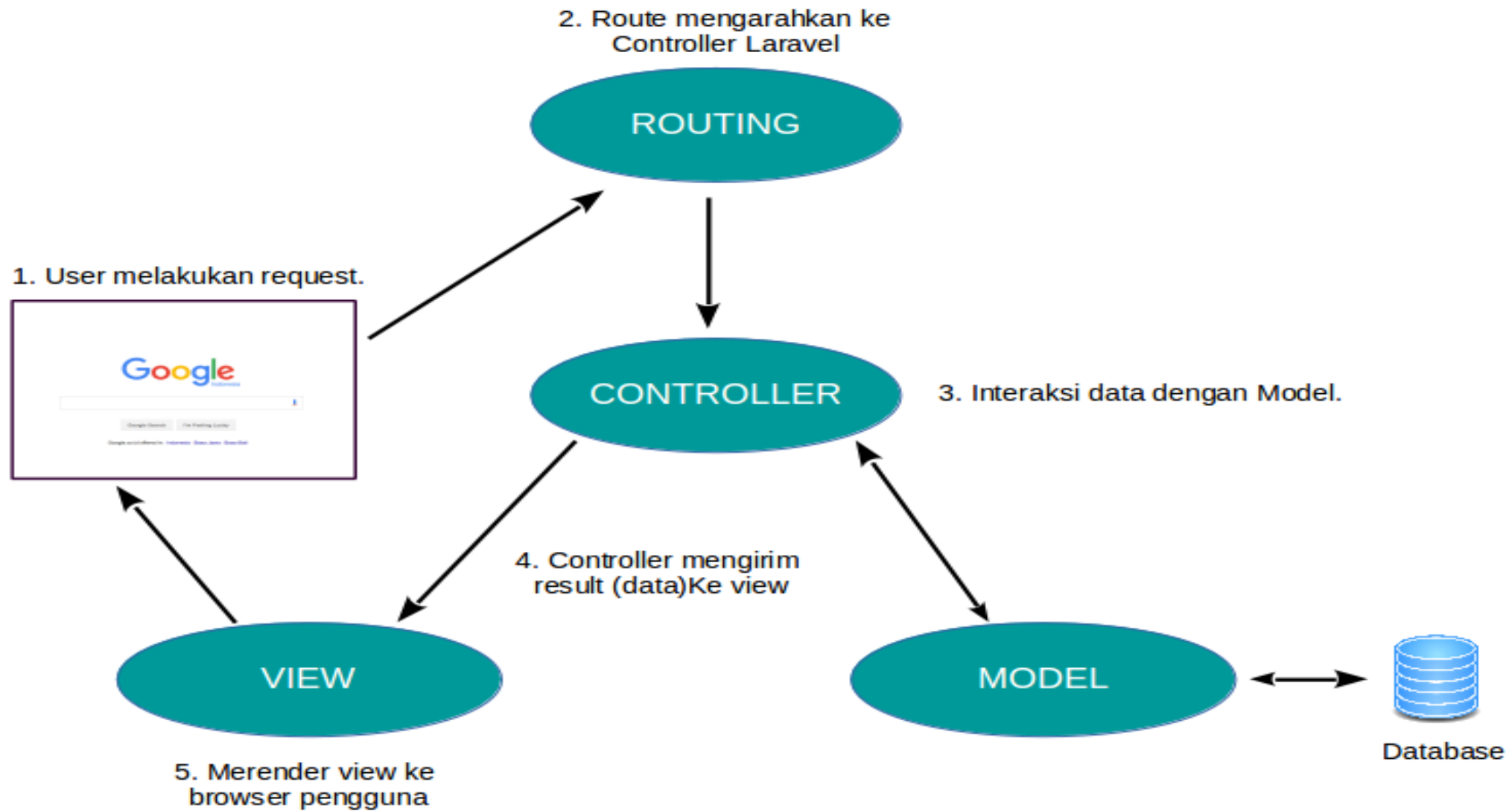


Söz dizimi çok basit ve anlamlıdır.



LARAVEL NASIL ÇALIŞIR?

Laravel, Model-View-Controller(MVC) adlı tasarım modeli kullanır.



LARAVELİN DOSYALAMA YAPISI

MIGRATIONS

```
php artisan make:migration create_name_table --create=TableName
```

- Migration, veritabanı mimarisinin sürüm kontrolünü sağlayarak, SQL tablolarını manuel oluşturmak yerine SQL' e gitmeden veritabanı işlemlerini gerçekleştirebileceğimiz bir ara katmandır.

```
public function up()
{
    Schema::create( table: 'Brand', function (Blueprint $table) {
        $table->id();
        $table->string( column: 'BrandName');
        $table->string( column: 'BrandCode');
        $table->string( column: 'Description');
        $table->string( column: 'Picture');
        $table->timestamps();
    });
}
```


.ENV DOSYASI

Bu dosyada uygulamanın çalışabilmesi için gerekli uygulama, veritabanı ve mail server gibi ayarlar bulunmaktadır.

```
DB_CONNECTION=pgsql  
DB_HOST=127.0.0.1  
DB_PORT=5432  
DB_DATABASE=project  
DB_USERNAME=postgres  
DB_PASSWORD=
```

```
Route::prefix( prefix: "brand")->group(function(){  
    Route::get( uri: "/index", action: "BrandController@index");  
    Route::post( uri: "/create", action: "BrandController@create");  
    Route::post( uri: "/update/{id}", action: "BrandController@update");  
    Route::delete( uri: "/delete/{id}", action: "BrandController@delete");  
});
```

ROUTİNG

Route ile sisteminize gelecek olan istekleri Controller, View gibi bölümlere yönlendir ve istek için çalışacak kod parçacıklarına ulaşırsınız.

- Controller katmanında view ve model arasında iletişim sağlanır. Modelden alınan dosyalar işlenerek view katmanına gönderilir.

CONTROLLER

php artisan make:controller ControllerName

MODEL

"Model", uygulamanızın üzerinde çalıştığı verilerin şeklini temsil eder. Kullanıcıların yaptığı postların bir listesini içeren bir kullanıcı tablonuz varsa, bu sizin modelinizdir.

```
php artisan make:model  
    ModelName
```

VIEWS

- Controller bu bilgiyi bir "**View**" oluşturmak için kullanır. View, görüntülenebildiği bir şablondur ve controller tarafından değiştirilebilir.

POSTMAN



- Postman, uzun kodlara ihtiyaç duymadan API'lerinizi kolayca test edebileceğiniz, API'nızın dokümentasyonunu hazırlamanızı sağlayan ve "rest client" olarak adlandırabileceğiniz bir araçtır.