Robust yet Efficient Web Development using Laravel

Alif Bhaskoro - 13514016
Computer Science Department
School of Electrical Engineering and Informatics
Institut Teknologi Bandung, Bandung, Indonesia
13514016@std.stei.itb.ac.id

***Abstract*—PHP has been used for web development for a quite long time. But plain PHP web development resulting in time consuming, inefficient, and vulnerable website. Laravel is a framework which standardized the development process, resulting robust scalability and improve developing efficiency.**

Keywords: Laravel, Web development, Robust, Efficient

I. INTRODUCTION

Nowadays, internet and website are common things. It's common to see many companies already have their own web-based system to support their business system. And this web technology is becoming more and more crucial for business development as time goes by. How to create a web-based system is not the main challenge, but the challenge is How to create a stable, scalable, efficient web-based system.

One of the most popular technique to build a web-based system is using PHP programming language (only) without using any web development framework. But this technique is way too simple, resulting in time consuming, inefficient, and vulnerable against web-based attacks (if developer doesn't handle it well)

This paper discuss another way to build a web-based system using laravel framework. Laravel is a PHP web development framework using model-view-controller (MVC) architectural pattern. Laravel provide key features such as eloquent ORM, query builder, blade templating engine, migrations, etc that make web-development more efficient than using the plain method (plain PHP programming). Laravel also have a default protection for common web-based attacks, such as SQL injections and cross-site request forgery (CRSF). This key features make laravel become a good way to develop a robust yet efficient website.

This paper is organized into five sections. The first section is introduction. The second section will explain some study about PHP programming language, and model-view-controller (MVC) architectural pattern. The third section will explain some key features that laravel have to create a robust yet efficient website. The fourth section will describe the experiment and discuss the result. And finally, the fifth section is the conclusion.

II. LITERATURE STUDY

A. PHP

PHP Hypertext Processor (PHP) is a server scripting language. PHP script can be injected between HTML codes, resulting a dynamic web pages.

PHP script starts with <?php and ends with ?>. This script can be written anywhere in the document. Every PHP statements end with semicolons (;).

```
<!DOCTYPE html>
<html>
<body>

<?php
    echo "Hello World!";
?>

</body>
</html>
```

Figure 1. Simple PHP script

PHP multi-line comment starts with /* and ends with */ or start with // for single-line comment. PHP variables start with dollar sign ($). PHP supports many data types, such as string, integer, float, boolean, float, etc. PHP commands are not case sensitive but PHP variables are.

```
<!DOCTYPE html>
<html>
<body>

<?php
    //This is single-line comment

    /* This is
    multi-line
    comment*/

    $i = 5;
?>

</body>
</html>
```

Figure 2. PHP script

B. MVC Architecture Pattern

Model-view-controller is a popular software architecture pattern for developing a website. It divided the application into three interconnected parts:

1. Model: Lowest level application part which is responsible for maintaining data.
2. View: Application part which is responsible for displaying data.
3. Controller: Application part which is controlling the interaction between Model and View.

MVC architecture pattern also defined the interaction between this three parts. Controller will send commands to model. Model will store data or retrieve data depends on commands from the controller, then will be passed to the view. A view will generate a new output for user based on the updated data from model.

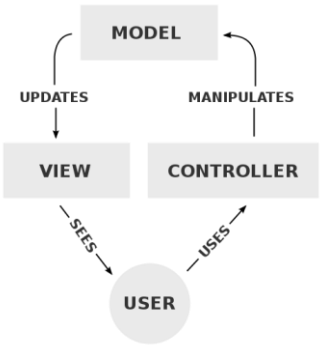

Figure 3. MVC interaction diagram

III. THE PROPOSED METHOD

This section explains about web development using laravel framework. We won't discuss all of laravel's feature, but we will discuss some of laravel's key feature. The features are Eloquent ORM, blade templating engine, migrations, and CRSF protection which will be described in each sub-section below.

A. Eloquent ORM

Laravel has its own default object relational mapping (ORM) called Eloquent. Eloquent is really useful to develop a website efficiently because it will simplify the code. To use this feature, we must create an Eloquent model first.

```php
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class LogAdmin extends Model
{
    //
    protected $table = 'tb_log_admin';
    public $timestamps = false;
    protected $primaryKey = 'id';
}
```

Figure 4. Eloquent Model

Model class will have 3 default variables that should be set when we created it. The first variable is table that should be set with the name of database table we want to map this model to. The second variable is timestamp. By default, eloquent expects to have created_at and updated_at columns on the tables. If we don't want to add this 2 columns, we can set timestamp variable with false. The third variable is primaryKey. By default, eloquent will assume the primary key name is 'id'. If we wish to change it, we can set the name of primary key to primaryKey variable.

To retrieve the value from database, Eloquent has methods which are equivalent with SQL queries. We will discuss some of them.

1. Get: equivalent with SQL query "Select * from tables"
2. Where('x', 'y'): equivalent with SQL query "Select * from tables where 'x'='y'"
3. First: equivalent with SQL query "limit by 1"
4. Save: equivalent with SQL query "Insert into tables" if primary key hasn't found in the table and "Update" if the primary key found in the table.

```php
//Select * from tb_log_admin
$arr = LogAdmin::get();

//print the id of each element of result
foreach($arr as $a){
    echo $a->id;
}
```

Figure 5. Eloquent usage

B. Blade Templating Engine

Laravel has its own default templating engine called Blade. File that use this templating engine will be saved as .blade.php instead of .php only. Blade has 2 key features that make developing a website more efficient which are:

1. Extends and yield, used to inject a content from child view into its parent view.
2. Control structures, shortcut for common PHP control structures. Ex: {{$arr}} is equivalent with <?php echo $arr; ?>.

C. Migrations

Laravel use migrations to maintain its database consistency. Migrations will create a new table in database called migrations. This table has 3 columns, id, migrations, and batch. Column id is a unique identifier. Column migrations contains the migration file name. Column batch contains the batch number which will be used for rollback.

The migration file has two functions, up and down. Function up will be called every time we use command php artisan migrate. It will check which file haven't migrated based on migration table in database and run their function up. To revert the changes, we can use command php artisan migrate:rollback. It will called every file with the biggest batch number in the table and called its down function.

```php
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class AddAdminActionHistory extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        //
        Schema::create('tb_log_admin', function (Blueprint $table) {
            $table->increments('id');
            $table->string('admin_username',32);
            $table->integer('transaction_id');
            $table->string('action',50);
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        //
        Schema::dropIfExists('tb_log_admin');
    }
}
```

Figure 6. Migration file

id	migration	batch
15	2017_02_15_034735_create_users_table	1
16	2017_02_15_084151_create_log_table	1
19	2017_02_20_114947_create_category_table	2
20	2017_02_20_120252_create_stuff_table	3
21	2017_02_20_125134_create_lend_table	4
46	2017_02_20_131044_create_transaction_table	5
47	2017_02_20_134004_create_payment_method_table	5

Figure 7. Migration table

D. *CRSF Protection*

By default, laravel force developer to add 1 input named _token to every php form. Every time form got submitted, server will check its input _token and compare it with token value that saved in session. If they are not matched, then the request is from other party and will be rejected because it could be a CRSF attack.

```php
<form action="{{ route('add_stuff') }}" method="post">
  <div class="form-group">
    <label for="name">Name:</label>
    <input type="test" name="name">
  </div>
  <input type="hidden" name="_token" value="{{ Session::token() }}">
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

Figure 8. Input _token in Laravel form

IV. Experiment Results

Before we start discussing the experiment results, imagine you and your team (3-4 people) had developed a website before and going to scale up that website.

The first problem you will encounter is how to modify your website's database schema on all of your team local environment. Of course you can send the newest SQL dump which contains all past data and will take around 10-15 minutes to be imported. Or you can just send the latest query to each members, but still it's not efficient. With migrations, you don't have to worry about modify database. You just need to use command php artisan migrate, and the system will run the newest query so your team database will be always consistent in local environment.

The second problem you will encounter is how to manipulate data in your database. With the plain method, you will need to open connection and run the query every time you want manipulate the database. It is inconvenient and inefficient. And what if you want to change the source table. With this plain method, you should change all the table name in all files. But with eloquent ORM, you only need to change variable table in the related model. And with using eloquent ORM, you don't need to open connection and make new query every time you want access the database. You just need to use eloquent method (get, where, etc.).

The third problem is imagine you have a navigation bar on 1000 web pages. If you use plain method, you will need to change all HTML codes in all 1000 files. With blade templating engine, you can use extends and yield feature to create consistent navigation bar. And to modify it you only need to change the navigation bar file.

```php
<body>
    <nav class="navbar navbar-default">
        <div class="container-fluid">
            <div class="navbar-header">
                <a class="navbar-brand" href="/all">Rentuff Admin</a>
            </div>
            <ul class="nav navbar-nav">
                <li class="dropdown">
                <a class="dropdown-toggle" data-toggle="dropdown" href="#">Transactions <span
                class="caret"></span></a>
                  <ul class="dropdown-menu">
                    <li><a href="/all">View All Transactions</a></li>
                    <li><a href="/admin_log">Log Admin</a></li>
                  </ul>
                </li>
            </ul>
        </div>
    </nav>

    <div class="container">
        @yield('content')
    </div>
</body>
```

Figure 9. Navigation bar file

```php
@extends('layouts.layout_admin')

@section('content')
  <br><br>
  <div style="display:inline-block;">
    <div style="display:inline-block; width:1035px;">
      <b><p style="font-size:28px">Add Stuff</p></b>
    </div>
    <div style="display:inline-block; width:100px;">
      <a href="/stuff_list" style="font-size: 20px;">back </a>
    </div>
  </div>
@endsection
```

Figure 10. Body file

The fourth problem is about security problem. Laravel has default protection for CRSF with input type hidden named _token. If the value doesn't match with the token value in server, request will be declined.

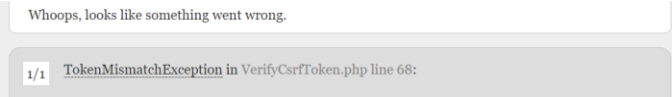

Figure 11. Token mismatch exception

V. CONCLUSION

With the plain method of web development using PHP programming language only resulting a vulnerable and inefficient web development. Using some key features of laravel like blade templating engine and eloquent ORM, we can easily build a robust web system more efficiently.

There are still many laravel features which author hasn't discussed in this paper yet like unit test, etc. Those features can be explored more and resulting a more efficient web development method.

ACKNOWLEDGMENT

Author would like to thank God because none of this work could be finished without His help. Author would like to thank Mr. Rinaldi Munir for his guidance and chance so this paper could be finished. And lastly author would like to thank everyone who helped in making this paper.

REFERENCES

[1] http://www.php.net/
[2] https://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm
[3] https://en.wikipedia.org/wiki/Model-view-controller

Bandung, 5 May 2017
Alif Bhaskoro
13514016