

# Laravel Main MVC Structure

## Main Modules

1. **view** → For all the pages to view.

**Location Example:** C:\xampp\htdocs\transport1\resources\views

2. **routes** → Maintain all the page link routing through **web.php** file

**Location Example:** C:\xampp\htdocs\transport1\routes\web.php

3. **Controllers** → All the functions are creating and maintaining here.

**Location Example:** C:\xampp\htdocs\transport1\app\Http\Controllers

Steps to create controller: (a) Open Terminal (**View → Tool Windows → Terminal**)

(b) Type the command: **php artisan make controller <controller name>Controller**

**Example: php artisan make controller FunctionsController**

**Location Example:** C:\xampp\htdocs\transport1\app\Http\Controllers\FunctionsController.php

4. **Model** → For creating and maintaining Data-Base table instance.

Steps to create controller: (a) Open Terminal (**View → Tool Windows → Terminal**)

(b) Type the command: **php artisan make model <Modelame>**

**Example: php artisan make controller VehicleInspConnector**

**Location Example:** C:\xampp\htdocs\transport1\app\VehicleInspConnector.php

## Work

### 1. Steps to create static page and view

- a. Create a static page under **view** folder or you can create your own folder under **view** folder with **.blade** extension

Page Name Example: welcome.blade.php

- b. To view a static page, you have to write some routing code in **web.php** page under **routes** folder.

Code: 

```
Route::get('<page name>', function () {  
    return view('<page name>');  
});
```

Example: 

```
Route::get('myhome', function () {  
    return view('myhome');  
});
```

### 2. Steps to create dynamic page and view

- a. Same as static page.

- b. To view a dynamic page you need to write the following code in **web.php** page.

Code:

```
Route::get('<pagename>', array('as'=>'<pagename>', 'uses'=>'<ControllerName>@<viewfunctionName>'));
```

Example:

```
Route::get('vehicleInspHome', array('as'=>'vehicleInspHome', 'uses'=>'FunctionsController@show'));
```

- c. As it's a dynamic page so you have to follow these steps to view the page.
  - i. Create a controller page first. (Just for One Time)
  - ii. Then under the controller class of controller page create a view function. (For every page create new view function in controller page)

Code:

```
class <ControllerName> extends Controller  
{  
    function <methodName>() {  
        return View::make('<path of the page>');  
    }  
}
```

Example:

```
class FunctionsController extends Controller  
{  
    function showCarInspDataEntry() {  
        return View::make('entryForm/carInspDataEntry');  
    }  
}
```

- d. To a link to view the page

`<a href="{{url('<link>')}}"></a>` Example: `<a href="{{url('<home>')}}"></a>`

### 3. Data Base Connection

Data base connection is very easy in Laravel. Steps are as follows.

- Open **.env** file (Location Example: C:\xampp\htdocs\transport1\.env)
- Edit the file to change the values of **DB\_DATABASE = <databaseName>**,  
**DB\_USERNAME = <username>**, **DB\_PASSWORD = <password>**
- And save the file

Example:

```
APP_ENV=local
APP_KEY=base64:s0uwrQpkgv+OaFrXVQZw77+/wAJri3MxG5EDcgRdFN4=
APP_DEBUG=true
APP_LOG_LEVEL=debug
APP_URL=http://localhost

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=vehicleinspdb
DB_USERNAME=root
DB_PASSWORD=
```

### 4. Access Data from Data Base

- To access the data of database a **model** file has to create.

Open Terminal and write the command: **php artisan make:model <model name>**

Example: **php artisan make:model VehicleInspConnector**

Location example: C:\xampp\htdocs\transport1\app\VehicleInspConnector.php

- Then create the instance of the data table which you want to access. To do this type the following command under the model class.

Command : **protected \$table="<Data base table name>;**

Example : **protected \$table="vehicle\_inspection";**

N.B: **\$table** variable is a system variable you must not change it.

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class VehicleInspConnector extends Model
{
    protected $table="vehicle_inspection";
}

?>
```

## 5. Insert data in to Data Base

- Create entry form page.
- Add the following code at the starting of <form> tag.

```
<?php echo Form::open(array('route' => '<insertFunctionName>', 'class' => 'form-control'));?>
```

Example:

```
<?php echo Form::open(array('route' => 'insertDataProducer', 'class' => 'form-control'));?>
```

- Add link in **web.php** for insert function routing.

Command:

```
Route::post('<insertFunctionName>', array('as'=>'<insertFunctionName>', 'uses' =>'<ControllerName>@<insertFunctionName>'));
```

Example:

```
Route::post('insertdata', array('as'=>'insertdata', 'uses'=>'FunctionsController@insertdata'));
```

- Go to the controller page.

Location Example: C:\xampp\htdocs\transport1\app\Http\Controllers\FunctionsController.php

- Create insert function.

Command:

```
public function insertdata(Request $request) {
    //print_r($request);
    //echo "insert data function called";
    //$data= Input::all();
    //print_r($data);

    //$request variable is used to receive Form Data

    $data = new VehicleInspConnector();//initialise database table object
    $data->insp_date = $request->date;
    $data->insp_time = $request->time;
    $data->vehicle_regno = $request->vehicleRegNo;
    $data->insp_no = $request->vehicleInspNo;
    $data->producer_name = $request->agentName;
    $data->save();

    return redirect()->route('vehicleInspHome'); // After saving data
    the command is used to redirect the page.
}
```

## 6. View Data

- Create a data view page with Edit & Delete option.
- Set routing link on **web.php** page.

Code:

```
Route::get('<pageName>', array('as'=>'<pageName>', 'uses'=>'<ControllerName>@<viewFunctionName>'));
```

Example:

```
Route::get('vehicleInspHome', array('as'=>'vehicleInspHome', 'uses'=>'FunctionsController@show'));
```

- Create a view function on controller page.

Code:

```
function <viewFunctionName>() {  
    $<variableName> = <ModelClassNameofDBtable>::all(); //to  
    fetch the data from database  
    return View::make('<path_of_view_page>')->with('<variable>', $<variable>);  
}
```

Example:

```
<?php  
namespace App\Http\Controllers;  
  
use App\Mydata;  
use App\producerConnector;  
use App\VehicleInspConnector;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\Input;  
use Illuminate\Support\Facades\Session;  
use Illuminate\Support\Facades\View;  
  
class FunctionsController extends Controller  
{  
  
    function show() {  
        $data = VehicleInspConnector::all();  
        return View::make('entryForm/vehicleInspHome')->with('mydata', $data);  
    }  
}  
?>
```

d. Write a @foreach function on view page to fetched data in view page.

Example:

```
<table class="table" id="table">
  <thead>
    <tr>
      <th class="text-center">Date</th>
      <th class="text-center">Time</th>
      <th class="text-center">Vehicle Reg No.</th>
      <th class="text-center">Inspection No.</th>
      <th class="text-center">Producer Name</th>
      <th class="text-center">Actions</th>
    </tr>
  </thead>
  <tbody>

    @foreach($mydata as $item)

      <tr>
        <!--<td>{{ $item->id }}</td> -->
        <td>{{ $item->insp_date }}</td>
        <td>{{ $item->insp_time }}</td>
        <td>{{ $item->vehicle_regno }}</td>
        <td>{{ $item->insp_no }}</td>
        <td>{{ $item->producer_name }}</td>

        <td>
          <a href="viewInspData/{{ $item->id }}" class="btn btn-success">
            <span class="glyphicon glyphicon-edit"></span> View
          </a>
          <a href="editInspData/{{ $item->id }}" class="btn btn-info">
            <span class="glyphicon glyphicon-edit"></span> Edit
          </a>
          <a href="deleteCustomer/{{ $item->id }}" class="btn btn-danger"
onclick="return ConfirmDelete()">
            <span class="glyphicon glyphicon-edit"></span> Delete
          </a>
        </td>
      </tr>
    @endforeach

  </tbody>
</table>
```

## 7. Edit Data

- Create a link in edit button and pass the **id** (Primary Key) of the that data which you want to edit.

Code Example:

```
<a href="editInspData/{{$item->id}}" class="btn btn-info"> <span class="glyphicon glyphicon-edit"></span> Edit</a>
```

- Set routing link in web.inf file

Code Example:

```
Route::get('editInspData/{editInspData_id}', array('as'=>'editInspData', 'uses'=>'FunctionsController@editInspDataView'));
```

- Create a function at controller page to fetch the data from DB and view the data through an edit page.

Code example of the function.

```
<?php
namespace App\Http\Controllers;

use App\Mydata;
use App\producerConnector;
use App\VehicleInspConnector;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Input;
use Illuminate\Support\Facades\Session;
use Illuminate\Support\Facades\View;

class FunctionsController extends Controller
{

    public function editInspDataView($editInspData_id) {
        //echo "Edit function called";
        //echo "Edit id is:". $editInspData_id;
        $edit_data= VehicleInspConnector::find($editInspData_id);
        return View::make('editForm/editCarInspData')->with('edit_insp_data', $edit_data);
    }

}
```

d. Create an edit page with update button. And add the following line of code in place of starting form tag.

```
<?php echo Form::open(array('url' => ['<link of update function >', $<DB variable >->id]));?>
```

Example Code of edit page

```
<html>
<h2>Vehicle Inspection Data Edit Form</h2>
<!-- <form> -->
<?php echo Form::open(array('url' => ['updateInspData', $edit_insp_data->id]));?>
<table>
  <tr>
    <td><label>Date:</label></td>
    <td><input type="text" name="date" value="{{ $edit_insp_data->insp_date }}"></td>
  </tr>
  <tr>
    <td><label>Time:</label></td>
    <td><input type="text" name="time" value="{{ $edit_insp_data->insp_time }}"></td>
  </tr>
  <tr>
    <td><label>Vehicle Reg No.:</label></td>
    <td><input type="text" name="vehicleRegNo" value="{{ $edit_insp_data->vehicle_regno }}"></td>
  </tr>
  <tr>
    <td><label>Vehicle Inspection No.:</label></td>
    <td><input type="text" name="vehicleInspNo" value="{{ $edit_insp_data->insp_no }}"></td>
  </tr>
  <tr>
    <td><label>Agent Name:</label></td>
    <td><input type="text" name="agentName" value="{{ $edit_insp_data->producer_name }}"></td>
  </tr>
  <tr><td></td></tr>
  <tr>
    <td><button type="submit" name="update">Update</button></td>
  </tr>
</table>
</form>
</html>
```

e. Create a update function at controller page.

Code Example:

```
public function updateInspData(Request $request, $updateInspData_id) {
    // echo "this is update function";
    $updateInspData = VehicleInspConnector::find($updateInspData_id);

    $updateInspData->insp_date = $request->date;
    $updateInspData->insp_time = $request->time;
    $updateInspData->vehicle_regno = $request->vehicleRegNo;
    $updateInspData->insp_no = $request->vehicleInspNo;
    $updateInspData->producer_name = $request->agentName;
    $updateInspData->save();

    return redirect()->route('vehicleInspHome');
```



*f. Set routing link on **web.php** page*

*Code Example:*

```
Route::post('updateInspData/{updateInspData_id}', array('as'=>'updateInspData', 'uses'=>'FunctionsController@updateInspData'));
```

## **8. Delete Data**