

CODE SCHOOL

*Presents*

# TRY LARAVEL

AN *INTERACTIVE* AND  
*EDUCATIONAL PRODUCTION*





Level 3 – Section 1

# Controllers & Routing

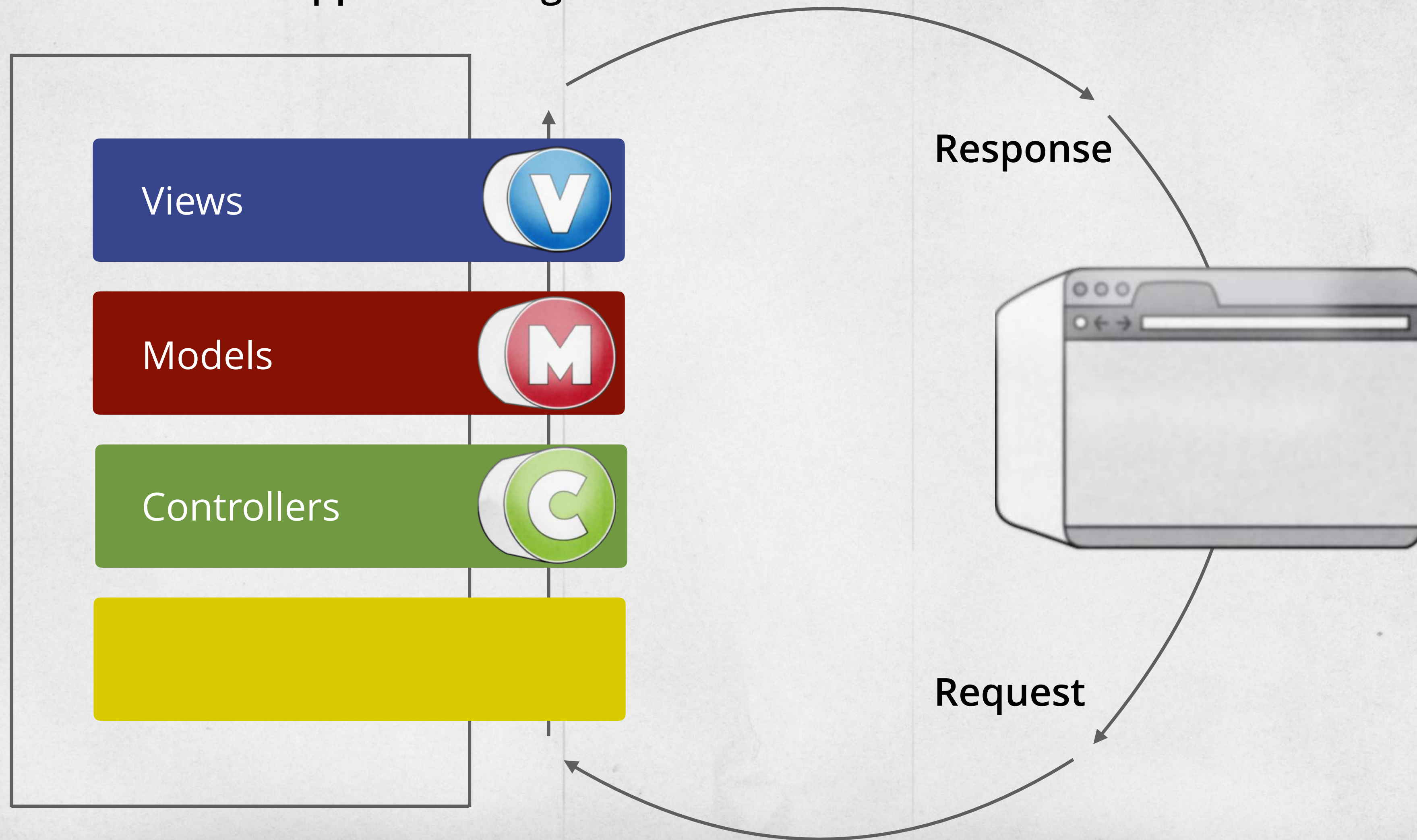
Controllers: The Middleman





# Application Stack Diagram

Controllers handle the application logic and render our views.



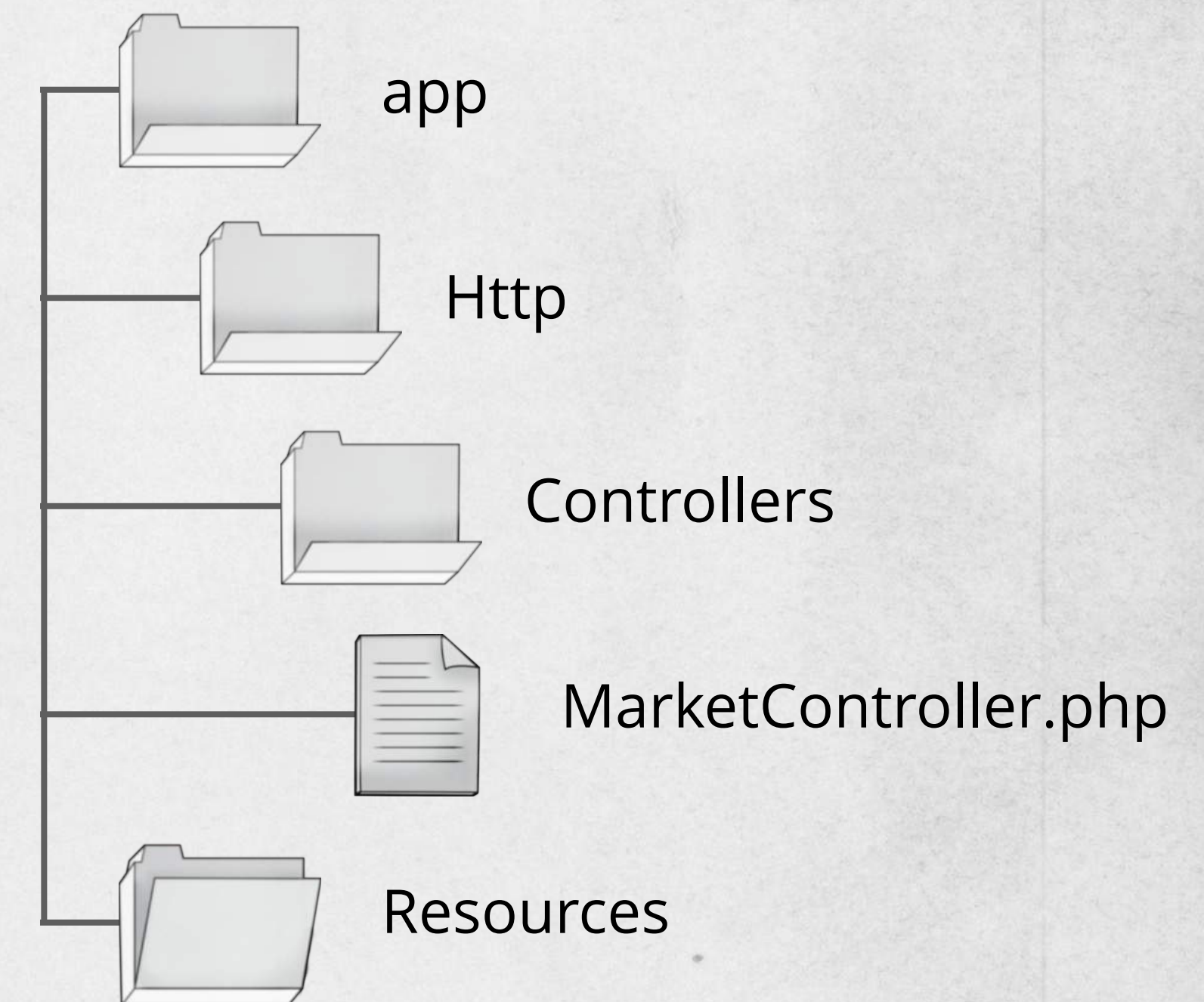


# Creating a New Controller



app/Http/Controllers/MarketController.php

```
namespace App\Http\Controllers;  
  
use App\Market;  
use Illuminate\Http\Request;  
  
class MarketController extends Controller  
{  
  
}  
}
```





# Moving Logic to the Controller



app/Http/Controllers/MarketController.php

```
namespace App\Http\Controllers;
```

```
use App\Market;
```

```
use Illuminate\Http\Request;
```

```
class MarketController extends Controller
```

```
{
```

```
    // Show All Markets
```

```
    public function index()
```

```
    {
```

```
    }
```

```
}
```

This action will be used to view all markets



# Moving Logic to the Controller



app/Http/Controllers/MarketController.php

```
namespace App\Http\Controllers;

use App\Market;
use Illuminate\Http\Request;

class MarketController extends Controller
{
    // Show All Markets
    public function index()
    {
        $markets = Market::all();
    }
}
```

Make a call to the Market model to get all markets!



# Moving Logic to the Controller



app/Http/Controllers/MarketController.php

```
namespace App\Http\Controllers;
```

```
use App\Market;
```

```
use Illuminate\Http\Request;
```

```
class MarketController extends Controller
```

```
{
```

```
    // Show All Markets
```

```
    public function index()
```

```
    {
```

```
        $markets = Market::all();
```

```
        return view('markets.index', ['markets' => $markets]);
```

```
    }
```

```
}
```

The first argument is our view template location

The second argument is used to pass along any data we want to the view!



# All Markets Using a Controller



Using the controller, we can now remove the call to query all markets.

## Resources/views/markets/index.blade.php

```
@extends( 'layouts.app' )
@section( 'main' )

@php
    $markets = Market::all();
@endphp

@foreach ( $markets as $market )
    <a href="{{ $market->site }}">
        <h2>{{ $market->name }}</h2>
    </a>
@endforeach
```

@php and @endphp in MVC  
should be a sign of code smell





# All Markets Using a Controller



Using the controller, we can now remove the call to query all markets.

## Resources/views/markets/index.blade.php

```
@extends('layouts.app')
@section('main')

@php
    $markets = Market::all();
@endphp

@foreach ($markets as $market)
    <a href="{{ $market->site }}">
        <h2>{{ $market->name }}</h2>
    </a>
@endforeach
```

Markets are now passed from  
the controller!





# Single Market Logic



To show a single market, we can create a new method in the controller called show.

app/Http/Controllers/MarketController.php

```
Class MarketController extends Controller
{
```

```
    // Show All Markets
```

```
    public function index()
```

```
    {
```

```
        $markets = Market::all();
```

```
        return view('markets.index', ['markets' => $markets]);
```

```
    }
```

```
    // Show Single Market
```

```
    public function show(Market $market)
```

```
    {
```

```
        return view('markets.show', ['market' => $market]);
```

```
    }
```

```
}
```

We are now passing an argument from the request to our method, the Market object



# Showing a Single Market



Again, we can now just remove the logic that is in the controller.

## Resources/views/markets/show.blade.php

```
@extends( 'layouts.app' )
@section( 'main' )

@php
    $market = Market::find(3);
@endphp

<h1>{{ $market->name }}</h1>
<h3>{{ $market->city }}</h3>
site:
<a href="{{ $market->site }}">
    {{ $market->site }}
</a>
```





# Showing a Single Market



Again, we can now just remove the logic that is in the controller.

## Resources/views/markets/show.blade.php

```
@extends('layouts.app')
@section('main')
```

```
@php
```

```
$market = Market::find(3);
```

```
@endphp
```

```
<h1>{{ $market->name }}</h1>
```

```
<h3>{{ $market->city }}</h3>
```

```
site:
```

```
<a href="{{ $market->site }}">
```

```
    {{ $market->site }}
```

```
</a>
```

Markets are now passed from  
the controller!



# Controllers in Review

---



Using controllers for logic

Using methods to render views

Removing logic from views

**TRY LARAVEL**



CODE SCHOOL

*Presents*

# TRY LARAVEL

AN *INTERACTIVE* AND  
*EDUCATIONAL PRODUCTION*





Level 3 – Section 2

# Controllers & Routing

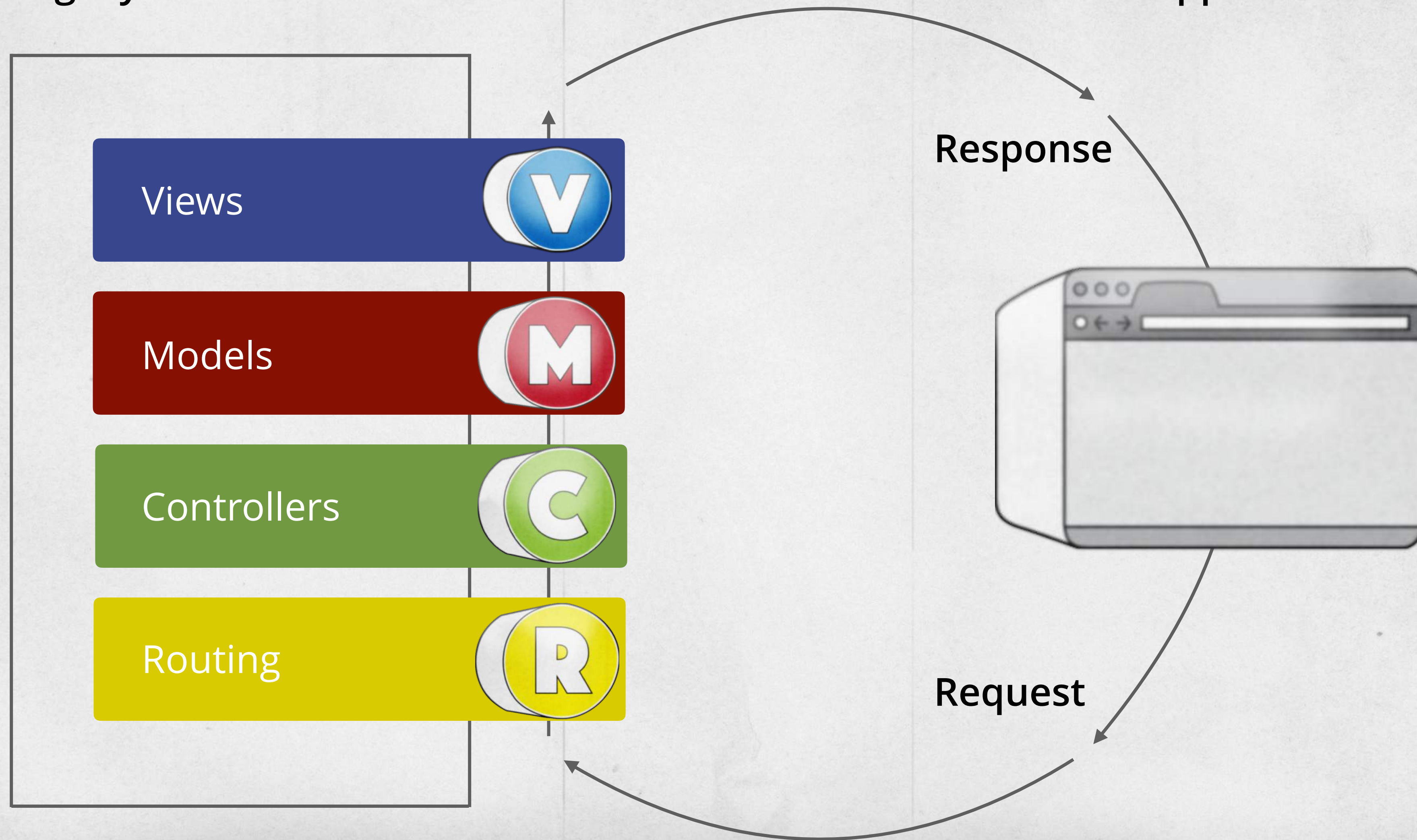
Routing: Directing Traffic





# Application Stack Diagram

The routing layer directs our traffic from the user-entered URL to our application.





# Routing, Starting Simple



For the most basic of routes, we will accept a location and then return a string.

routes/web.php

```
<?php
```

```
Route::get('/', function () {  
    return 'Hello World';  
});
```

The first argument is the URL we are trying to match

```
Route::get('markets/{id}', function ($id) {  
    return 'Requested Market id = ' . $id;  
});
```

The second argument is a callback, or the code that will run if the URL is matched



# Routing, With Controllers



Since we are using controllers, we will map routes to controller actions.

routes/web.php

```
<?php
```

```
Route::get('/', 'MarketController@index');  
Route::get('markets', 'MarketController@index');
```

app/Http/Controllers/MarketController.php

```
Class MarketController extends Controller  
{  
    // Show All Markets  
    public function index()  
    {  
  
    }  
}
```



# Routing, With Controllers



Since we are using controllers, we will map routes to controller actions.

routes/web.php

```
<?php
```

```
Route::get('markets/create', 'MarketController@create');
```

```
Route::post('markets', 'MarketController@store');
```

```
Route::get('markets/{market}', 'MarketController@show');
```

market.dev/markets/3

market.dev/markets/create



# Routing, With Controllers



Since we are using controllers, we will map routes to controller actions.

routes/web.php

```
<?php
```

```
Route::get('markets/{market}/edit', 'MarketController@edit');
```

```
Route::patch('markets/{market}', 'MarketController@update');
```

```
Route::delete('markets/{market}', 'MarketController@destroy');
```

① market.dev/markets/3/edit



# Routing: One Line, Many Routes



With the one line, we now have many different routes!

routes/web.php

```
<?php
```

```
Route::get('/', 'MarketController@index');  
Route::resource('markets', 'MarketController');
```

Method	URI	Name	Action
GET	markets	markets.index	MarketController@index
GET	markets/create	markets.create	MarketController@create
POST	markets	markets.store	MarketController@store
DELETE	markets/{market}	markets.destroy	MarketController@destroy
PUT / PATCH	markets/{market}	markets.update	MarketController@update
GET	markets/{market}	markets.show	MarketController@show
GET	markets/{market}/edit	markets.edit	MarketController@edit



# Routing in Review

---



Routing with closures

CRUD routes

Resource routing for models and controllers

**TRY LARAVEL**



CODE SCHOOL

*Presents*

# TRY LARAVEL

AN *INTERACTIVE* AND  
*EDUCATIONAL PRODUCTION*

