Web Programming with The Laravel Framework

#5 - Routing

Steven Lolong

steven.lolong@uni-tuebingen.de

Programming Language Research Group
Tuebingen University

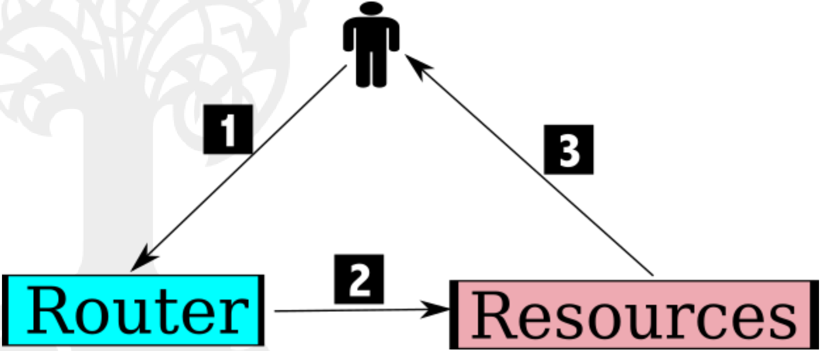
11 May 2022

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Routing

Routing, is a pattern in software design to protect the vital resource(s) from direct access of users.

How it works



How it works

1. Request from user, ex. sending information (view with web-form), triger from link in view or address to router.
2. Router will process (usually the target address) and map the request to resource.
3. Resource response the request from user

How it works

Let's construct a simple application that protects a vital file from direct access!

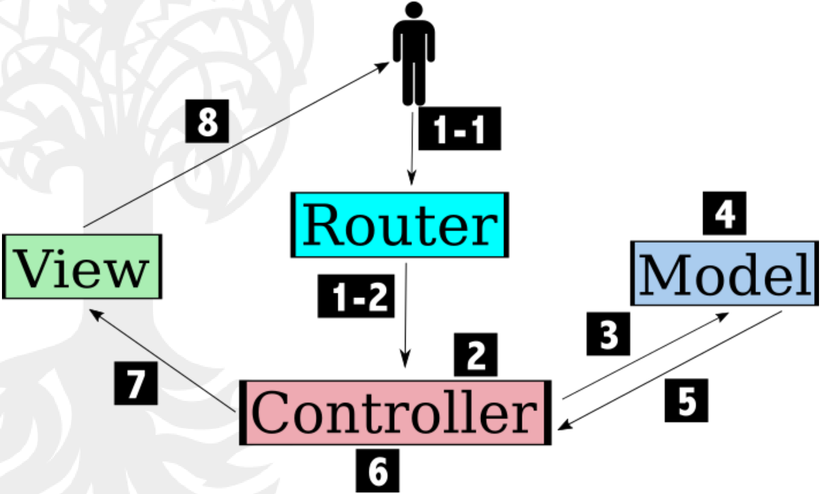
1. create a public folder, and create *router.php* inside.
2. set root document folder of web server to folder public.
3. create a controller folder, and create *controller.php* inside.

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

MVC++

MVC++ = **MVC** + **Routing**

How it works



How it works

1. Request
    1. Request from user, ex. sending information (view with web-form), triger from link in view or address to controller.
    2. Map the request to Controller.
2. Processing data in controller.
3. C-Request, it can be get or update data.
4. M-Process, processing C-Request in model.
5. M-Response, sending information to controller.
6. C-Process, processing information.
7. C-Update, update view.
8. V-Response, display UI to user.

Protecting the Vital Resources

Let's design the application using the ~~M~~ VC++.

1. create a public folder, and create *router.php* inside.
2. create a controller folder, and create *controller.php* inside.
3. create a view folder, and create *home.html, contact.html, and 404.html* inside.
4. create a helper folder (optional), and create *helper.php* inside.
5. create a config folder (optional), and create *global_env.php* inside.

Eberhard Karls Universität Tübingen

Router

```php
include "../config/global_env.php";
include "../controller/controller.php";


if(isset($_GET["routing"])){
switch($_GET["routing"]){
  case "home":
    showHome();
  break;
  case "contact":
    showContact();
  break;
  default:
    show404();
}
}else{
  showHome();
}
```

Listing: router – public/router.php

Controller

```php
<?php
include "../helper/address.php";
function showHome(){
  require_once "../view/home.html";
}

function showContact(){
  include "../view/contact.html";
}

function show404(){
  include "../view/404.html";
}

?>
```

Listing: model – controller/controller.php

View - Home

```
<html>

<head>
  <title>Home</title>
</head>

<body>
  <H1>Home</H1>
  <br />
  <a href="<?php echo rootAddress(); ?>home">Home</a> <br />
  <a href="<?php echo rootAddress(); ?>contact">Contact</a>
</body>

</html>
```

Listing: home – view/home.html

View - Contact

```
<html>

<head>
  <title>Contact</title>
</head>

<body>
  <H1>Contact Me</H1>
  <br />
  <a href="<?php echo rootAddress(); ?>home">Home</a> <br />
  <a href="<?php echo rootAddress(); ?>contact">Contact</a>
</body>

</html>
```

Listing: contact – view/contact.html

View - 404

```
<html>

<head>
  <title>404</title>
</head>

<body>
  <H1>404 - File not found!</H1>
  <br />
  <a href="<?php echo rootAddress(); ?>home">Home</a> <br />
  <a href="<?php echo rootAddress(); ?>contact">Contact</a>
</body>

</html>
```

Listing: 404 – view/404.html

Helper

```php
<?php
function rootAddress(){
  return $GLOBALS["webAddress"] . "/" . $GLOBALS["indexFile"] . "?routing=";
}

?>
```

Listing: helper – helper/address.php

Configuration File

```php
<?php
$GLOBALS["webAddress"] = "http://localhost";
$GLOBALS["indexFile"] = "router.php";
?>
```

Listing: global environment – config/global_env.php