

[Home](#) > [PHP](#) > [Creating a simple PHP routing with parameters.](#)

Creating a simple PHP routing with parameters.

[PHP](#) Yousuf — September 12, 2021 · 2 Comments

PHP is a general-purpose language that can be used to develop a range of different types of web applications. It has been traditionally used to create dynamic and interactive websites and web applications.

Routing :

Routing is a common term in web development. Its purpose is to tell your application how to handle incoming requests. It decides which function needs to process the request. Today we will see how we can make a simple routing in PHP.

PHP routing is an alternative to the default PHP mechanism for handling requests. It offers a very simple and intuitive way of defining routes and mapping them to specific functions. It is also a toolkit that helps you to make your web application more modular and maintainable.

Requirements :

- PHP 5+.
- Basic knowledge of PHP OOP.
- Basic knowledge of Regex.
- XAMPP knowledge.

Our first step for PHP routing is to create a **.htaccess** file that will send all the requests to our index.php file. I am assuming that you know XAMPP. Go to htdocs folder, create a new folder "routing" and create a new file **".htaccess"**. Add this code into your **.htaccess** file.

```
1 RewriteEngine On
2
3 #location of your website
4 RewriteBase /routing
5
6 RewriteCond %{REQUEST_FILENAME} !-f
7 RewriteCond %{REQUEST_FILENAME} !-d
8
9 RewriteRule ^(.+)$ index.php?url=$1 [QSA,L]
```

PHP Routing

PHP Routing Without Parameters :

Create "**index.php**" file then create a new "**Route**" class and add the public method "**add**".

```
<?php

class Route {

    /*
     * two parameters.
     *
     * route : route address
     * file : location of the file to show if route address matched
     */
    public function add($route, $file){

    }

}
```

Our next step is to create a "**home.php**" file at the same folder then go to "**index.php**" and create a new instance of **Route** and call the "**add**" method with the route address and file location.

```
<!-- home.php file -->

<h1>Home Page </h1>
```

```
<?php

//index.php file

class Route {

    /*
    two parameters.

    route : route address
    file : location of the file to show if route address matched

    */
    public function add($route, $file){

    }

}

//Route instance
$route = new Route();

//route address and home.php file location
$route->add("/home", "home.php")

?>
```

PHP has a global `$_REQUEST` variable, which can be used to get the URI of the request. The URI of the request is available in `$_REQUEST['uri']`. So by using this global variable we can match if the URL address is the same as the route address or not. We also r to remove start and end slashes from the URL address. See the below code you will understand better.

[Doc on \\$_REQUEST global variable](#)

```
public function add($route, $file){

    //replacing first and last forward slashes
    //$_REQUEST['uri'] will be empty if req uri is /

    if(!empty($_REQUEST['uri'])){
        $route = preg_replace("/(^\/)|(\\/$)/", "", $route);
        $reqUri = preg_replace("/(^\/)|(\\/$)/", "", $_REQUEST['uri']);
    }else{
        $reqUri = "/";
    }

    if($reqUri == $route ){

        //on match include the file.
        include($file);

        //exit because route address matched.
        exit();
    }

}
```

Now your code should look like this.

```

1  <?php
2
3  //index.php file
4
5  class Route {
6
7      /*
8      two parameters.
9
10     route : route address
11     file : location of the file to show if route address matched
12
13     */
14     public function add($route, $file){
15
16
17         //replacing first and last forward slashes
18         //$_REQUEST['uri'] will be empty if req uri is /
19
20         if(!empty($_REQUEST['uri'])){
21             $route = preg_replace("/^(^\/)|(\$/)/","",$route);
22             $reqUri = preg_replace("/^(^\/)|(\$/)/","",$_REQUEST['uri']);
23         }else{
24             $reqUri = "/";
25         }
26
27         if($reqUri == $route ){
28
29             //on match include the file.
30             include($file);
31
32             //exit because route address matched.
33             exit();
34         }
35
36     }
37 }
38
39 //Route instance
40 $route = new Route();
41
42 //route address and home.php file location
43 $route->add("home", "home.php");
44
45
46
47 ?>
48

```

I am assuming that you are using XAMPP. When you go to **http://localhost/routing/home/** you will see the contents of your home.php file. In the same way, you can add more routes. You can also create functions to control post/get requests easily in the located file.

```

$route->add("/home", "home.php");

$route->add("/user", "user.php");

$route->add("/about-us", "about.php");

```

What if user goto a route which is not defined by you? I mean 404 page.

To show a 404 page when no route is matched, you need to create a new method in the **"Route"** class, name it **"notFound"**. Call this method with your 404-page file location at the end in your **"index.php"** file.

```
public function notFound($file){
    include($file);
    exit();
}
```

```
1  <?php
2
3  //index.php file
4
5  class Route {
6
7      /*
8      two parameters.
9
10     route : route address
11     file : location of the file to show if route address matched
12
13     */
14     public function add($route, $file){
15
16         //replacing first and last forward slashes
17         //$_REQUEST['uri'] will be empty if req uri is /
18
19         if(!empty($_REQUEST['uri'])){
20             $route = preg_replace("/^(^\/)|(\$/)/","",$route);
21             $reqUri = preg_replace("/^(^\/)|(\$/)/","",$_REQUEST['uri']);
22         }else{
23             $reqUri = "/";
24         }
25
26         if($reqUri == $route ){
27
28             //on match include the file.
29             include($file);
30
31             //exit because route address matched.
32             exit();
33         }
34     }
35
36 }
37
38 public function notFound($file){
39     include($file);
40     exit();
41 }
42 }
43
44 //Route instance
45 $route = new Route();
46
47 //route address and home.php file location
48 $route->add("/home", "home.php");
49
50 $route->add("/user", "user.php");
51
52 $route->add("/about-us", "about.php");
53
54 //write it at the last
55 //arg is 404 file location
56
```

```

57 | $route->notFound("404.php");
58 |
    | ?>

```

PHP Routing With Parameters

We have learned basic routing in PHP, but what if we want to get a parameter from the route address? for example I want to access a user's id so my route should be like this: **"user/id"**. Now the question is how can I access the **id** ?.

This can be achieved and now I will explain you all the things which are required to get parameters from the route address.

STEP 1:

Remove everything from the **Route "add"** method.

```

public function add($route, $file){

}

```

STEP 2:

Create a new private method **"simpleRoute"** in the **Route** class. This will be called when there is no parameter required in the route address.

```

private function simpleRoute($file, $route){

    //replacing first and last forward slashes
    //$_REQUEST['uri'] will be empty if req uri is /

    if(!empty($_REQUEST['uri'])){
        $route = preg_replace("/(^\/)|(\\/$)/", "", $route);
        $reqUri = preg_replace("/(^\/)|(\\/$)/", "", $_REQUEST['uri']);
    }else{
        $reqUri = "/";
    }

    if($reqUri == $route){
        $params = [];
        include($file);
        exit();
    }

}

```

STEP 3:

In the **Route "add"** method add this code. Read the comments, I have explained each line. Basic knowledge of Regular expression is required to understand this.

```

1 | function add($route,$file){
2 |
3 |     //will store all the parameters value in this array
4 |     $params = [];
5 |
6 |

```

```

6 //will store all the parameters names in this array
7 $paramKey = [];
8
9 //finding if there is any {?} parameter in $route
10 preg_match_all("/(?<=){}.+?(?=})/", $route, $paramMatches);
11
12 //if the $route does not contain any param call simpleRoute();
13 if(empty($paramMatches[0])){
14     $this->simpleRoute($file,$route);
15     return;
16 }
17
18 //setting parameters names
19 foreach($paramMatches[0] as $key){
20     $paramKey[] = $key;
21 }
22
23
24 //replacing first and last forward slashes
25 //$_REQUEST['uri'] will be empty if req uri is /
26
27 if(!empty($_REQUEST['uri'])){
28     $route = preg_replace("/(^\/)|(\$/)/","",$route);
29     $reqUri = preg_replace("/(^\/)|(\$/)/","",$_REQUEST['uri']);
30 }else{
31     $reqUri = "/";
32 }
33
34 //exploding route address
35 $uri = explode("/", $route);
36
37 //will store index number where {?} parameter is required in the $route
38 $indexNum = [];
39
40 //storing index number, where {?} parameter is required with the help of regex
41 foreach($uri as $index => $param){
42     if(preg_match("/{.*}/", $param)){
43         $indexNum[] = $index;
44     }
45 }
46
47 //exploding request uri string to array to get ^
48 //the exact index number value of parameter from $_REQUEST['uri']
49 $reqUri = explode("/", $reqUri);
50
51 //running for each loop to set the exact index number with reg expression
52 //this will help in matching route
53 foreach($indexNum as $key => $index){
54
55     //in case if req uri with param index is empty then return
56     //because url is not valid for this route
57     if(empty($reqUri[$index])){
58         return;
59     }
60
61     //setting params with params names
62     $params[$paramKey[$key]] = $reqUri[$index];
63
64     //this is to create a regex for comparing route address
65     $reqUri[$index] = "{.*}";
66 }
67
68 //converting array to sting
69 $reqUri = implode("/", $reqUri);
70
71

```

```

72         //replace all / with \ for reg expression
73         //regex to match route is ready !
74         $reqUri = str_replace("/", '\\', $reqUri);
75
76         //now matching route with regex
77         if(preg_match("/$reqUri/", $route))
78         {
79             include($file);
80             exit();
81         }
82     }

```

Now we can add routes like this to get the parameter.

```
$route->add("/user/{id}", "user.php");
```

In the “**user.php**” file you can access id like this :

```

<?php

//user.php file

<h1> echo $params['id'] </h1>

?>

```

Similarly, you can get multiple parameters.

```
$route->add("/user/{id}/{username}/{country}", "user.php");
```

```

<?php

//user.php file

<h1> echo $params['id'] </h1>
<h1> echo $params['username'] </h1>
<h1> echo $params['country'] </h1>

?>

```

This is the final code :

```

1  <?php
2
3  class Route {
4
5      private function simpleRoute($file, $route){
6
7
8          //replacing first and last forward slashes
9          //$_REQUEST['uri'] will be empty if req uri is /
10
11         if(!empty($_REQUEST['uri'])){
12             $route = preg_replace("/(^\\)|(\\$)/", "", $route);
13             $reqUri = preg_replace("/(^\\)|(\\$)/", "", $_REQUEST['uri']);
14         }else{
15             $reqUri = "/";
16

```



```
17     }
18
19     if($reqUri == $route){
20         $params = [];
21         include($file);
22         exit();
23     }
24 }
25
26 }
27
28 function add($route,$file){
29
30     //will store all the parameters value in this array
31     $params = [];
32
33     //will store all the parameters names in this array
34     $paramKey = [];
35
36     //finding if there is any {?} parameter in $route
37     preg_match_all("/(?<={}).+?(?={})/", $route, $paramMatches);
38
39     //if the route does not contain any param call simpleRoute();
40     if(empty($paramMatches[0])){
41         $this->simpleRoute($file,$route);
42         return;
43     }
44
45     //setting parameters names
46     foreach($paramMatches[0] as $key){
47         $paramKey[] = $key;
48     }
49
50
51     //replacing first and last forward slashes
52     //$_REQUEST['uri'] will be empty if req uri is /
53
54     if(!empty($_REQUEST['uri'])){
55         $route = preg_replace("/^(^\\)|(^\\$)/","",$route);
56         $reqUri = preg_replace("/^(^\\)|(^\\$)/","",$_REQUEST['uri']);
57     }else{
58         $reqUri = "/";
59     }
60
61     //exploding route address
62     $uri = explode("/", $route);
63
64     //will store index number where {?} parameter is required in the $route
65     $indexNum = [];
66
67     //storing index number, where {?} parameter is required with the help of regex
68     foreach($uri as $index => $param){
69         if(preg_match("/{.*}/", $param)){
70             $indexNum[] = $index;
71         }
72     }
73
74     //exploding request uri string to array to get
75     //the exact index number value of parameter from $_REQUEST['uri']
76     $reqUri = explode("/", $reqUri);
77
78     //running for each loop to set the exact index number with reg expression
79     //this will help in matching route
80     foreach($indexNum as $key => $index){
81
```

```

82         //in case if req uri with param index is empty then return
83         //because url is not valid for this route
84         if(empty($reqUri[$index])){
85             return;
86         }
87
88         //setting params with params names
89         $params[$paramKey[$key]] = $reqUri[$index];
90
91         //this is to create a regex for comparing route address
92         $reqUri[$index] = "{.*}";
93     }
94
95     //converting array to sting
96     $reqUri = implode("/", $reqUri);
97
98     //replace all / with \ for reg expression
99     //regex to match route is ready !
100    $reqUri = str_replace("/", '\\', $reqUri);
101
102    //now matching route with regex
103    if(preg_match("/$reqUri/", $route))
104    {
105        include($file);
106        exit();
107    }
108
109 }
110
111 function notFound($file){
112     include($file);
113     exit();
114 }
115 }
116
117 $route = new Route();
118
119 $route->add("/user/{id}", "user.php");
120
121 //example route with multiple params
122 $route->add("/download/{downloadID}/{filename}", "download.php");
123
124 $route->notFound("404.php");
125
126 ?>

```

This is how you can create simple PHP routing and access parameters. I hope you learned something from this article, if you have any issues or questions feel free to comment below I will be happy to answer.

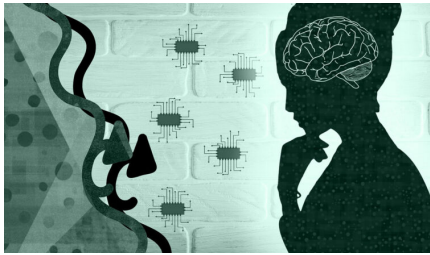
READ : [Send emails with Laravel using PHPMailer](#)

Github Repo: <https://github.com/yousufnoor5/php-basic-routing>



Yousuf

Related Posts



Simple Machine Learning in PHP

2 Comments



Marcin March 22, 2022 at 6:33 pm

if I use `$route->add("/user/{id}","user.php")` how do I can to return to download? it looks like "user" directory and I can't return previous files



★ **helpincoding** March 22, 2022 at 6:38 pm

Hi, You can go back by using `"../"`.

Example : `$route->add("/user/{id}", "../download.php")`

You can read more about how to go back :

<https://stackoverflow.com/questions/162873/how-do-i-include-a-file-over-2-directories-back>

COMMENTS ARE CLOSED.

[Privacy Policy](#) | [About](#) | [Contact](#)

© 2023 HelpInCoding

