MY PHP FRAMEWORK DOCUMENTATION

v0.8.3

Contents

Views 3

How to create view 3

Variables 3

Including views 4

CSRF 5

Models 6

Controllers 7

Routing 9

Views

How to create view

All view files are located in views folder (app/views). View files must have .view.php extension. Middlewear allows you to use few syntactic sugars in your view, such as printing variables, including other views etc...

Example I: home view

File: **home.view.php**

```
<h1>Test</h1>

<br>
```

Variables

Printing variables in views is also possible. (variables are defined in controllers).

For printing variables you have to use *{{var}}*.

Example II: home view with variables

File: **home.view.php**

```
<h1>Test</h1>

<br>

My name is: {{ir['username']}}
```

Including views

Example III: home view with more views included

Only 1 level of included files is supported. Included files can not include more files.

File: **home.view.php**

```
[include]app/views/base.view.php[/include]

<h1>Test</h1>

<br>

My name is: {{ir['username']}}

[include]app/views/footer.view.php[/include]
```

CSRF

Framework offers basic CSRF protection. You cang et CSRF token value in view using variable CSRF .

If you want to include CSRF field in form, use CSRFFORM variable.

Example IV: Using CSRF token in views

File: **home.view.php**

```
[include]app/views/base.view.php[/include]
<h1>Test</h1>
<br>
My name is: {{ir['username']}}<br>
<br>

Reading from cache: {{cVal}} <br>
CSRF TOKEN: {{CSRF}} <br>
<form action='' method='post'>
{{CSRFFORM}}<br>
Username: <input type='text' name='username'>
<br>
<input type='submit'>
</form>
[include]app/views/footer.view.php[/include]
```

Models

Controllers

All controllers extend main controller called Controller.

Your controller contains 2 functions, *post()* and *get()*. If user makes post request, *post()* function is called, if user makes get request then *get()* function is called.

You can show view with two different functions: *show()* or *render()*.

- show() function shows view without any additional functions.
- *render()* function shows view + uses cache and caches view after you change it. It also uses gzip compression.

Example V: Custom controller

```
<?php
require_once 'Controller.php';
class Home extends Controller{
        public function __construct(){
        }
        public function post() {
        }
        public function get() {
                $this->render("home.view.php");
        }
}
```

Example VI: Showing view using show function

```
<?php
        $this->show("home.view.php");
```

Example VII: Showing view using render function

```
<?php

        $this->render("home.view.php");
```

If you want view to show variables, you can pass them to view, using additional argument.

Example VIII: Passing data to view

```
<?php

$reloadTimer = 5;

$moreInfo = "Hello";

$data = ["reloadTimer" => $reloadTimer,

          "extraInformation" => $moreInfo];

$this->render("home.view.php", $data);

// or $this->show("home.view.php", $data);
```

Routing

Routing through pages is easy. All you have to do is to add one line in **index.php** file.

```
Router::home('Main', 'app/controllers/Main.php');
```

If you want to set default/home page use function home(<page name> , <controller).

```
Router::make('Main', 'app/controllers/Main.php');
```

At the end you must call function route(). This function navigates through pages. Navigation through pages is done using page GET parameter ($_GET['page']).

Example VI: Routing

File: **index.php**

```
Router::home('Main', 'app/controllers/Main.php');

Router::make('Main', 'app/controllers/Main.php');

Router::route();
```