

Лабораторна робота №19

Створення системи маршрутизації

Хід роботи

```

1  <?php
2
3  class Router
4  {
5      private $routes = [];
6
7      // match method handler
8      public function addRoute($method, $path, $handler)
9      {
10         $this->routes[] = [
11             'method' => $method,
12             'path' => $path,
13             'handler' => $handler,
14         ];
15     }
16
17     // matchMethod method handler handlerArray
18     public function match($method, $uri)
19     {
20         foreach ($this->routes as $route) {
21             // matchMethod type key key keyArray method key keyArray key key
22             $pattern = str_replace('/', '%2F', $route['path']);
23             $pattern = preg_replace('/\A(.*?)\Z/','(?<bb>^/$)/', $pattern);
24             $pattern = '/^' . $pattern . '$/';
25
26             if ($route['method'] == $method && preg_match($pattern, $uri, $matches)) {
27                 // matchMethod key keyArray route
28                 $route['params'] = array_filter($matches, 'is_string', ARRAY_FILTER_USE_KEY);
29                 return $route;
30             }
31         }
32
33         return null;
34     }
35
36     // matchMethod key
37     public function handleRequest()
38     {
39         $method = $_SERVER['REQUEST_METHOD'];
40         $uri = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);
41
42         $route = $this->match($method, $uri);
43
44         if ($route) {
45             if (is_callable($route['handler'])) {
46                 // key keyArray key keyArray key keyArray keyArray
47                 call_user_func_array($route['handler'], $route['params'] ?? []);
48             } else {
49                 // keyArray key keyArray keyArray key key key key key key
50                 $class = $route['handler'][0];
51                 $pdo = $route['handler'][1];
52                 $method = $route['handler'][2];
53
54                 // keyArray keyArray keyArray
55                 $params = $_REQUEST;
56
57                 // key key POST key keyArray key key key key
58                 if ($method === 'POST') {
59                     $postData = file_get_contents('php://input');
60                     $postData = json_decode($postData, true);
61
62                     if ($postData != null) {
63                         $params = array_merge($params, $postData);
64                     }
65                 }
66
67                 $classInstance = new $class($pdo);
68                 // keyArray key key keyArray keyArray
69                 call_user_func_array([$classInstance, $method], $params);
70             }
71         } else {
72             // keyArray key key keyArray keyArray key
73             header('HTTP/1.1 404 Not Found');
74             echo '404 Not Found';
75         }
76     }
77 }
78

```

```

1 K:\ppp
2 class MaterialController extends Controller {
3     public function index() {
4         $materialsModel = new Material($this->pdo, "materials");
5         $materials = $materialsModel->all();
6         $this->renderMaterials($materials);
7     }
8
9     public function show($id) {
10         $materialsModel = new Material($this->pdo, "materials");
11         $material = $materialsModel->find($id);
12         $this->renderMaterial($material);
13     }
14
15     protected function renderMaterials($materials) {
16         echo "<h3>Material list</h3>";
17         echo "<table border='1'>";
18         echo "<tr><th>ID</th><th>Product</th><th>Amount</th></tr>";
19
20         foreach ($materials as $material) {
21             echo "<tr>";
22             echo "<td> . $material['id'] . "</td>";
23             echo "<td> . $material['product'] . "</td>";
24             echo "<td> . $material['amount'] . "</td>";
25             echo "</tr>";
26         }
27
28         echo "</table>";
29     }
30
31     protected function renderMaterial($material) {
32         echo "<h3>Material Details</h3>";
33         echo "<table border='1'>";
34         echo "<tr><th>ID</th><th>Product</th><th>Amount</th></tr>";
35
36         if ($material != null) {
37             echo "<tr>";
38             echo "<td> . $material['id'] . "</td>";
39             echo "<td> . $material['product'] . "</td>";
40             echo "<td> . $material['amount'] . "</td>";
41             echo "</tr>";
42         } else {
43             echo "<tr><td colspan='3'>Material not found</td></tr>";
44         }
45
46         echo "</table>";
47     }
48 }
49 }>

```

					Лабораторна робота № 19						
Зм	Лист	№ докум	Підпис	Дата	Створення системи маршрутизації				Літ.	Лист.	Листів.
Розробив	Убоженко										
Перевірів	Левицький										
Оцінка											
Затв											
					Група 451						

Контрольні питання

1. Що таке маршрутизація в контексті веб-розробки?
Маршрутизація - це процес визначення та обробки URL-шляхів у веб-додатках, який спрощує розподіл трафіку до відповідних обробників.
2. Які основні елементи маршрутизації в PHP-роутері, який ти надав, і що вони роблять?
URL-шляхи та відповідні обробники (контролери).
HTTP-методи (GET, POST, тощо) для різних дій.
3. Як можна визначити новий маршрут у роутері? Які параметри приймає цей метод?
`$router->addRoute('/example', 'ExampleController@index', 'GET');`
4. Що таке параметри маршруту, і як вони використовуються в роутері?
Параметри маршруту визначаються, наприклад, так: `/user/{id}`. Вони використовуються для передачі змінних значень у URL.
5. Чому важливо мати підтримку різних HTTP-методів у маршрутизації?
Різні HTTP-методи (GET, POST, PUT, DELETE) використовуються для виконання різних дій на сервері. Підтримка різних методів дозволяє обробляти різні види запитів.
6. Які переваги може надати використання класів-контролерів в роутері замість безіменних функцій?
Класи-контролери забезпечують структурований підхід та можливість групування пов'язаних функцій у класах, що полегшує управління кодом.
7. Що робить метод `handleRequest` у роутері, і чому він важливий?
Метод `handleRequest` обробляє вхідний HTTP-запит, визначає відповідний маршрут та викликає відповідний обробник (контролер).
8. Як можна передати параметри конструктору класу контролера через роутер?
Використовуйте функцію `addRoute` так: `$router->addRoute('/example/{id}', 'ExampleController@index', 'GET', ['id' => '\d+'])`;
9. Як можна вдосконалити цей роутер для обробки винятків, таких як помилка 404?
Додайте перевірку і обробку помилок, наприклад, перенаправлення на сторінку 404 при відсутності відповідного маршруту.
10. Що робить функція `addRoute` у роутері, і які аргументи вона приймає?
Функція `addRoute` додає новий маршрут до роутера. Вона може приймати URL-шлях, контролер, HTTP-метод і, за потреби, параметри для обмежень.

					<i>Лабораторна робота № 19</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		