

[Головна](#) [Вхід](#) [Реєстрація](#)

Каталог бази будівельних матеріалів

[Головна](#) [Вхід](#) [Реєстрація](#)

Реєстрація

Логін:

Пароль:

Повторіть пароль:



Зареєструватися

[← Вхід](#)

[Головна](#) [Вхід](#) [Реєстрація](#)

Вхід

Логін:

Пароль:

Увійти

[← Реєстрація](#)

Будівельні матеріали

| ID | Назва | Категорія | Постачальник | Ціна | Одиниця | Дія |
|----|-------|-----------|--------------|--------|---------|---|
| 9 | 123 | 123 | 123 | 222.00 | un | Переглянути Редагувати Видалити |

[Додати матеріал](#)

Категорії матеріалів

[Додати категорію](#)

| ID | Назва | Дія |
|----|-------|---|
| 7 | 123 | Редагувати Видалити |

Додати категорію

Назва:

[← Назад до списку](#)

Постачальники

[Додати постачальника](#)

| ID | Назва | Телефон | Email | Дія |
|----|-------|---------|---------------|---|
| 7 | 123 | 222 | sdjfsdj@sdkad | Переглянути Редагувати Видалити |

Додати постачальника

Назва:

Телефон:

Email:

[← Назад до списку](#)

Матеріал: 123

Категорія: 123

Постачальник: 123

Ціна: 222.00

Одиниця виміру: un

[← Назад до списку](#) [Редагувати](#) [Видалити](#)

Редагувати матеріал

Назва:

Категорія:

Постачальник:

Ціна:

Одиниця:

[← Назад](#)

1. Що таке маршрутизація в контексті веб-розробки?

Маршрутизація — це процес, який визначає, який код виконати, коли користувач заходить на певний URL.

2. Основні елементи маршрутизації в PHP-роутері

У простому PHP-роутері зазвичай є такі ключові елементи:

Маршрут (Route) — це зв'язок між URL і дією (callback або контролером).

Методи HTTP — GET, POST, PUT, DELETE тощо; вони визначають, який тип запиту обробляти.

Контролер або callback — функція чи метод класу, який виконується, коли маршрут спрацьовує.

Параметри маршруту — динамічні частини URL, які передаються в контролер.

3. Як визначити новий маршрут у роутері?

У більшості роутерів є метод типу `addRoute` або окремі методи для HTTP-методів:

```
$router->addRoute('GET', '/users/{id}', [UserController::class, 'show']);
```

Параметри методу `addRoute` зазвичай такі:

Метод HTTP (GET, POST і т.д.)

Шлях маршруту (`/users/{id}`)

Дія — callback або масив [Клас, метод]

4. Що таке параметри маршруту і як їх використовують?

Параметри маршруту — це змінні частини URL.

Наприклад: `/users/{id}` — тут `{id}` буде динамічним.

Якщо користувач заходить на `/users/5`, роутер витягує `id = 5` і передає його контролеру:

```
public function show($id) {  
    echo "User ID: $id";  
}
```

5. Чому важливо мати підтримку різних HTTP-методів?

Різні методи відповідають різним типам дій:

GET — отримати дані

POST — створити нові дані

PUT — оновити дані

DELETE — видалити

Якщо роутер не розрізняє методи, ти не зможеш правильно розділити дії для одного URL.

6. Переваги використання класів-контролерів

- Легше організовувати код.
- Можна використовувати спільні методи та властивості (наприклад, доступ до моделі чи бази).
- Код стає чистішим і тестованим.
- Можна підключати middleware для контролера.
- Без класів все доводиться робити через безіменні функції, що швидко стає хаотичним.

7. Що робить метод `handleRequest` у роутері?

`handleRequest` аналізує запит користувача:

Визначає URL і метод HTTP.

Шукає маршрут, який підходить під цей URL і метод.

Викликає відповідну функцію або метод контролера.

Це серце роутера, без нього маршрути не працюють.

8. Як передати параметри конструктору контролера через роутер?

Якщо контролер очікує параметри в конструкторі, можна створювати його через анонімну функцію:

```
$router->addRoute('GET', '/users/{id}', function($id) {  
    $controller = new UserController($someDependency);  
    $controller->show($id);  
});
```

Тобто роутер викликає функцію, а вона вже створює контролер з потрібними параметрами.

9. Як вдосконалити роутер для обробки винятків (404, 500)?

404 — коли маршрут не знайдено:

```
if (!$routeFound) {  
    http_response_code(404);  
    echo "Сторінку не знайдено";  
}
```

500 — для помилок всередині контролера, можна обгорнути в `try/catch`.

Можна додати окремий метод `setNotFoundHandler()` для зручності.

10. Що робить функція `addRoute`?

Вона додає новий маршрут у масив маршрутизатора.

Аргументи:

Метод HTTP (GET, POST)

URL-шлях (`/users/{id}`)

Callback або контролер

Після цього роутер знає, що робити, коли користувач заходить на цей URL.