

Event Driven Programming

With PHP Framework

Berikut adalah beberapa library/framework PHP untuk event-driven programming:

1. Symfony Event Dispatcher - untuk aplikasi berbasis Symfony.
2. Laravel Events - sistem event bawaan Laravel.
3. EventSauce - untuk event sourcing.
4. League Event - solusi ringan untuk event management.
5. ReactPHP - untuk aplikasi asinkron dan real-time.
6. Zend/Laminas Event Manager - untuk aplikasi berbasis Laminas.

Pilihannya tergantung pada kebutuhan aplikasi Anda:

- Gunakan **Laravel** atau **Symfony** jika sudah menggunakan framework tersebut.
- Gunakan **ReactPHP** untuk aplikasi real-time.
- Gunakan **League Event** untuk solusi yang sederhana dan ringan.

1. Symfony Event Dispatcher

Symfony adalah framework PHP yang sangat populer, dan salah satu komponennya, **EventDispatcher**, dirancang untuk mendukung event-driven programming.

Cara Kerja:

- Anda dapat membuat event dan listener.
- Listener terdaftar untuk event tertentu dan dijalankan saat event dipicu.

Contoh :

```
use Symfony\Component\EventDispatcher\EventDispatcher;
use Symfony\Component\EventDispatcher\Event;

require 'vendor/autoload.php';

// Definisikan Event
class UserRegisteredEvent extends Event {
    public $username;

    public function __construct($username) {
        $this->username = $username;
    }
}

// Listener
function sendWelcomeEmail(UserRegisteredEvent $event) {
    echo "Welcome email sent to " . $event->username . "\n";
}

// Dispatcher
$dispatcher = new EventDispatcher();
$dispatcher->addListener('user.registered', 'sendWelcomeEmail');

// Memicu Event
```

```
$event = new UserRegisteredEvent('john_doe');  
$dispatcher->dispatch($event, 'user.registered');
```

Kegunaan:

- Digunakan di banyak aplikasi berbasis Symfony, termasuk Laravel (yang meminjam beberapa ide dari Symfony).

2. Laravel Events

Laravel memiliki sistem event bawaan yang sangat kuat, yang memungkinkan penerapan event-driven programming dengan mudah.

Cara Kerja:

- Anda membuat event dan listener.
- Event dipicu, dan listener yang terdaftar dijalankan.

Contoh:

1. Buat Event:

php artisan make:event UserRegistered

File event ini akan muncul di app/Events/UserRegistered.php:

```
namespace App\Events;  
  
class UserRegistered {  
    public $username;  
  
    public function __construct($username) {  
        $this->username = $username;  
    }  
}
```

2. Buat Listener:

php artisan make:listener SendWelcomeEmail

File listener ini akan muncul di app/Listeners/SendWelcomeEmail.php:

```
namespace App\Listeners;  
  
use App\Events\UserRegistered;  
  
class SendWelcomeEmail {
```

```
        public function handle(UserRegistered $event) {  
            echo "Sending welcome email to " . $event->username;  
        }  
    }  
}
```

3. Daftarkan Listener:

Tambahkan di app/Providers/EventServiceProvider.php:

```
protected $listen = [  
    UserRegistered::class => [  
        SendWelcomeEmail::class,  
    ],  
];
```

4. Emit Event:

```
use App\Events\UserRegistered;  
  
event(new UserRegistered('john_doe'));
```

Laravel mempermudah pengelolaan event dan listener dengan berbagai fitur tambahan, seperti **queue** untuk event asynchronous.

3. ReactPHP

[ReactPHP](#) adalah framework PHP untuk aplikasi asinkron yang berbasis event-driven programming. Ini sering digunakan untuk aplikasi real-time, seperti server WebSocket atau chat.

Cara Kerja:

- Semua operasi, seperti membaca file atau menangani request, dipicu oleh event.

Contoh :

```
require 'vendor/autoload.php';

$loop = React\EventLoop\Factory::create();
$loop->addTimer(2, function () {
    echo "This is executed after 2 seconds\n";
});

echo "Waiting...\n";
$loop->run();
```

Kegunaan:

- Cocok untuk aplikasi real-time seperti server WebSocket, streaming data, atau aplikasi asinkron lainnya.