



Laravel Logging

Eko Kurniawan Khannedy

Eko Kurniawan Khannedy

- Technical architect at one of the biggest ecommerce company in Indonesia
- 11+ years experiences
- www.programmerzamannow.com
- youtube.com/c/ProgrammerZamanNow





Eko Kurniawan Khannedy

- Telegram : [@khannedy](https://t.me/khannedy)
- Facebook : fb.com/ProgrammerZamanNow
- Instagram : instagram.com/programmerzamannow
- Youtube : youtube.com/c/ProgrammerZamanNow
- Telegram Channel : t.me/ProgrammerZamanNow
- Email : echo.khannedy@gmail.com



Sebelum Belajar

- Laravel Dasar
- PHP Logging



Agenda

- Pengenalan Laravel Logging
- Logging Configuration
- Log Facade
- Context
- Handler
- Formatter
- Dan lain-lain

Pengenalan Laravel Logging



Pengenalan Logging

- Materi Logging sudah kita bahas tuntas di kelas PHP Logging



Pengenalan Laravel Logging

- Laravel sendiri tidak membuat fitur logging dari awal. Laravel menggunakan library monolog untuk melakukan logging
- Oleh karena itu wajib menyelesaikan kelas PHP Logging, karena di kelas ini kita tidak akan bahas lagi library Monolog
- Walaupun Laravel menggunakan library Monolog untuk melakukan logging, namun Laravel juga menambah fitur untuk mempermudah penggunaan library Monolog, sehingga kita tidak perlu melakukan konfigurasi secara manual

Membuat Project



Membuat Project

- `composer create-project laravel/laravel=version nama-folder`
- <https://packagist.org/packages/laravel/laravel>

Logging Configuration



Logging Configuration

- Laravel secara default menggunakan file `config/logging.php` sebagai konfigurasi untuk logging nya
- Saat kita membuat project, secara default, sudah disediakan banyak sekali konfigurasi yang bisa kita gunakan, namun yang secara default aktif adalah yang sederhana



Logging Channel

- Laravel menyediakan beberapa channel (tujuan) log yang bisa kita gunakan, seperti :
- single, mengirim data log ke single file
- daily, mengirim data log ke single file, namun tiap hari akan di rotate file nya
- slack, mengirim data log ke slack chat
- syslog, mengirim data log ke syslog
- null, tidak mengirim data log kemanapun
- stack, mengirim data log ke beberapa channel sekaligus, default nya hanya mengirim ke channel single
- Secara default, Laravel akan menggunakan channel stack

Log Facade



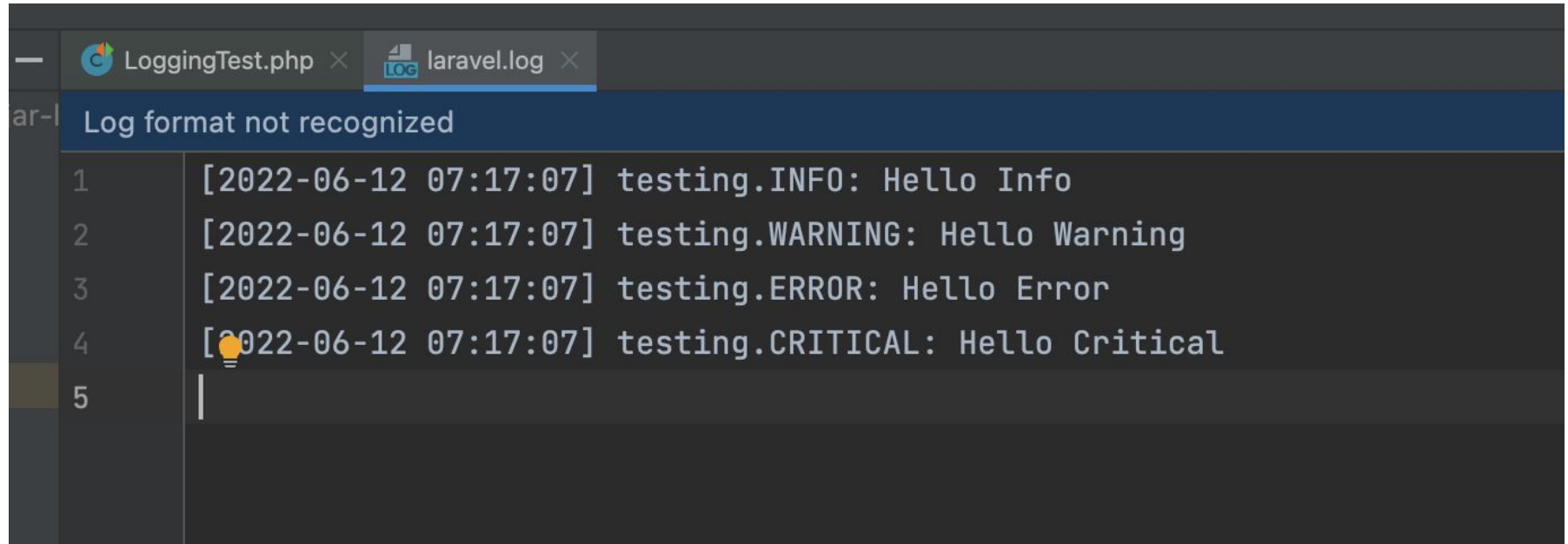
Log Facade

- Untuk melakukan logging di Laravel, kita bisa dengan mudah menggunakan Log Facade
- kita tidak perlu manual membuat Logger Monolog lagi
- <https://laravel.com/api/9.x/Illuminate/Support/Facades/Log.html>

Kode : Log Facade

```
5 use Illuminate\Support\Facades\Log;  
6 use Tests\TestCase;  
7  
8 class LoggingTest extends TestCase  
9 {  
10     public function testLog()  
11     {  
12         Log::info("Hello Info");  
13         Log::warning("Hello Warning");  
14         Log::error("Hello Error");  
15         Log::critical("Hello Critical");  
16  
17         self::assertTrue(true);  
18     }  
}
```


Kode : Hasil Log



```
LoggingTest.php x laravel.log x
Log format not recognized
1 [2022-06-12 07:17:07] testing.INFO: Hello Info
2 [2022-06-12 07:17:07] testing.WARNING: Hello Warning
3 [2022-06-12 07:17:07] testing.ERROR: Hello Error
4 [2022-06-12 07:17:07] testing.CRITICAL: Hello Critical
5 |
```

Multiple Log Channel



Multiple Log Channel

- Seperti yang sudah kita bahas di materi Logging Configuration, kita bisa menggunakan stack untuk mengirim data log ke beberapa channel
- Contohnya, kita bisa mengirim log menggunakan stack channel ke single, stderr dan slack sekaligus



Membuat Slack Webhook URL

- Untuk mengirim log ke slack, kita perlu membuat slack webhook URL terlebih dahulu
- <https://slack.com/apps/A0F7XDUAZ-incoming-webhooks>




Kode : Logging Configuration


```
'channels' => [  
  'stack' => [  
    'driver' => 'stack',  
    'channels' => ['single', 'stderr', 'slack'],  
    'ignore_exceptions' => false,  
  ],  
  
  'single' => [  

```



Hasil Slack

 added an integration to this channel: [incoming-webhook](#)

 **Laravel Log** APP 2:18 PM

Message
Hello Error
Level
ERROR
Laravel Log | Today at 2:18 PM

2:18 **Message**
Hello Critical
Level
CRITICAL
Laravel Log | Today at 2:18 PM

Bottom navigation bar with icons for Home, Search, Add, and others.

Context




Context

- Kita tahu saat belajar di PHP Logging, di Monolog terdapat fitur bernama Context
- Itu juga bisa kita gunakan di Laravel Logging
- Log Facade memiliki parameter kedua setelah message yang bisa kita isi dengan data context, mirip seperti yang kita lakukan ketika belajar PHP Logging menggunakan Monolog




Kode : Context



```
public function testContext()
{
    Log::info("Hello Context", ["user" => "khannedy"]);

    self::assertTrue(true);
}
```







With Context

- Atau, kita bisa gunakan function `withContext()`, yang secara otomatis kode selanjutnya akan menggunakan context yang kita gunakan dalam `withContext()`
- Ini sangat cocok ketika misal di Controller kita menambahkan `withContext()` berisi data user, sehingga di kode-kode selanjutnya, setiap log akan berisikan informasi siapa user yang melakukan proses logging tersebut

Kode : With Context

```
26
27   public function testWithContext()
28 {
29     Log::withContext(["user" => "khannedy"]);
30
31     Log::info("Hello Info");
32     Log::warning("Hello Warning");
33
34     self::assertTrue(true);
35  }
36
```



Selected Channel



Selected Channel

- Secara default, Laravel akan menggunakan logging channel DEFAULT yang sudah dipilih
- Namun, pada kasus tertentu, kita mungkin ingin membuat channel, tapi hanya digunakan ketika kita mau gunakan
- Laravel juga memiliki fitur menseleksi channel yang ingin digunakan, sehingga sebelum kita kirim data log, kita bisa pilih channel mana yang akan kita gunakan
- Kita bisa menggunakan method `channel(string)`
- Return dari method `channel()` adalah Logger, sehingga kita harus menggunakan Logger tersebut untuk mengirim ke channel yang kita pilih

Kode : Selected Channel

```
public function testWithChannel()
{
    $slackLogger = Log::channel('slack');
    $slackLogger->error("Hello Slack");

    Log::info("Hello Laravel"); // mengirim ke default channel

    self::assertTrue(true);
}
```

Handler



Handler

- Saat menggunakan driver monolog, kita bisa menentukan attribute handler yang berisi class Monolog Handler
- Contoh sebelumnya terdapat driver single untuk menyimpan data log ke file, sebenarnya jika kita mau, kita bisa menggunakan driver monolog dan handler StreamHandler



Kode : File Handler

```
'file' => [  
  'driver' => 'monolog',  
  'level' => env('LOG_LEVEL', 'debug'),  
  'handler' => StreamHandler::class,  
  'formatter' => env('LOG_STDERR_FORMATTER'),  
  'with' => [  
    'stream' => storage_path('logs/application.log'),  
  ],  
],
```



Kode : Test File Handler

```
47 public function testFileHandler()  
48 {  
49     $fileLogger = Log::channel('file');  
50     $fileLogger->info("Hello World");  
51     $fileLogger->warning("Hello World");  
52     $fileLogger->error("Hello World");  
53  
54     self::assertTrue(true);  
55 }  
56
```

Formatter



Formatter

- Saat kita menggunakan driver monolog, secara default akan menggunakan LineFormatter
- Pada kasus tertentu, misal kita ingin bentuk log dibuat dalam bentuk JSON misalnya
- Di Laravel, untuk melakukan itu, kita bisa gunakan config formatter dengan berisi class Formatter untuk monolog nya



Kode : Formatter

```
'file' => [  
  'driver' => 'monolog',  
  'level' => env('LOG_LEVEL', 'debug'),  
  'handler' => StreamHandler::class,  
  'formatter' => \Monolog\Formatter\JsonFormatter::class,  
  'with' => [  
    'stream' => storage_path('logs/application.log'),  
  ],  
],
```



Kode : Hasil Log

```
[2022-06-12 07:42:10] testing.INFO: Hello World  
[2022-06-12 07:42:10] testing.WARNING: Hello World  
[2022-06-12 07:42:10] testing.ERROR: Hello World  
{"message":"Hello World","context":{},"level":200,"level_name":"INFO","channel":"testing","datetime":"2022-06-12 07:42:10"}  
{"message":"Hello World","context":{},"level":300,"level_name":"WARNING","channel":"testing","datetime":"2022-06-12 07:42:10"}  
{"message":"Hello World","context":{},"level":400,"level_name":"ERROR","channel":"testing","datetime":"2022-06-12 07:42:10"}
```

Materi Selanjutnya



Materi Selanjutnya

- Laravel Blade Template
- Laravel Database / Eloquent
- Laravel Validation
- Laravel Command
- Laravel HTTP Client
- Dan lain-lain