

laravel daily log permission 문제 해결

- [개요](#)
- [artisan 권한 문제](#)
- [daily log 문제](#)
- [추천 방법](#)
 - [5.6 이상 - log channel 사용](#)
 - [5.5 이하](#)
- [참고](#)

개요

라라벨의 로그 설정은 `config/app.php` 의 다음 항목을 통해서 정의되며 변경은 `.env` 에 하면 됩니다.

```
'log' => env('APP_LOG', 'single'),  
'log_level' => env('APP_LOG_LEVEL', 'debug'),
```

기본적으로 로그는 `storage/logs/laravel.log` 에 남으며 로그 레벨은 debug 입니다.

로그와 session, view cache 등이 저장되는 storage 디렉터리와 bootstrap 디렉터리는 laravel 을 구동하는 웹 서버가 읽고 쓸 수 있도록 설정해야 합니다. 일반적으로 우분투는 www-data, CentOS 는 nginx 나 apache 계정을 사용해야 합니다.

ubuntu

```
$ sudo chown -R www-data:www-data storage bootstrap/cache
```

CentOS

```
$ sudo chown -R nginx:nginx storage bootstrap/cache
```

artisan 권한 문제

위와 같이 소유자를 웹 서버로 설정할 경우 artisan 같이 사용자 계정으로 구동하는 경우(job scheduler 포함) log 파일에 쓰기 권한이 없기 때문에 정상적으로 동작하지 않습니다.

이 문제를 해결하려면 다른 사용자도 로그 파일에 쓸수 있도록 권한을 777 로 수정해야 합니다.

```
$ sudo chmod -R 777 storage/logs
```

또는 access control 기능을 사용하여 artisan 을 실행할 사용자 계정(예: lesstif)이 로그 파일을 쓸 수 있도록 설정합니다.

```
$ sudo setfacl -R -m u:lesstif:rx storage/logs
```

이제 웹 서버에서나 커맨드에서 artisan 을 실행할때나 모두 laravel.log 을 쓸 수 있게 되어 문제가 해결되었습니다.

daily log 문제

운영 서비스라면 laravel.log 파일에 모든 로그를 기록하는 single log 설정은 로그 파일이 너무 커지게 되므로 검색이 어려워지고 운영이 불편해져서 바람직하지 않으므로 날짜별로 로그 파일을 rolling 하는 daily 설정을 하는게 좋습니다.

config/app.php

```
'log' => env('APP_LOG', 'daily'),
```

위와 같이 설정하면 *laravel-Y-m-d.log*(예: *laravel-2017-08-04.log*) 형식으로 날마다 로그 파일이 기록되므로 단일 로그 파일로 인한 문제를 해결할 수 있습니다.

하지만 날자가 바뀌어서 로그 파일이 새로 생성되는 경우 **laravel**은 기본 권한을 소유자(로그 파일을 새로 생성하는 프로세스, 보통은 웹 서버)만 읽고 쓸 수 있는 **0644**로 생성하므로 **artisan** 명령을 실행할 경우 로그 파일에 쓸 수 없어서 오작동하는 새로운 문제를 야기합니다.

이를 해결하려면 라라벨 프레임워크의 일부인 *Illuminate\Log\Writer.php* 파일을 수정해서 새로 생성되는 로그 파일의 권한이 777이 되도록 해야 합니다.

Illuminate\Log\Writer.php

```
public function useDailyFiles($path, $days = 0, $level = 'debug')
{
    $this->monolog->pushHandler(
        // $handler = new RotatingFileHandler($path, $days, $this->parseLevel($level))
        $handler = new RotatingFileHandler($path, $days, $this->parseLevel($level), true, 0777)
    );

    $handler->setFormatter($this->getDefaultFormatter());
}
```

하지만 이는 라라벨 프레임워크의 수정이므로 프레임워크를 업데이트했을 경우 매번 변경이 필요하므로 권장할 만한 방법이 아닙니다.

추천 방법

제가 추천하는 방법은 웹 서비스시 생성하는 로그 파일과 **artisan** 같이 커맨드에서 사용하는 로그 파일을 분리하는 것이며 **laravel** 버전따라 아래와 같이 설정하면 됩니다.

5.6 이상 - log channel 사용

이를 위해 *config/logging.php* 파일에 아래와 같이 **cli** 용 로그 채널을 추가하고 상단에는 **console** 실행 여부에 따라 **channel** 을 분리하는 코드를 추가합니다.

```

<?php

use Monolog\Handler\StreamHandler;
use Monolog\Handler\SyslogUdpHandler;

//+++++++
// console
$log_channel = env('LOG_CHANNEL', 'stack');
if ($app->runningInConsole()) {
    $log_channel = 'cli';
}
//+++++++

return [
    'default' => $log_channel, //

    'channels' => [
        'stack' => [
            'driver' => 'stack',
            // cli
            'channels' => ['daily'],
        ],
        // console channel
        'cli' => [
            'driver' => 'daily',
            'path' => storage_path('logs/laravel-cli.log'),
            'level' => 'debug',
            'days' => 14,
        ],
    ],
];

```

5.5 이하

5.5 이하는 *bootstrap/app.php* 에 custom log 를 설정해 줍니다.

이제 daily log 로 설정해도 웹 서비스의 로그 파일은 기본 설정인 *laravel-2017-08-03.log* 등으로 생성되며 artisan 로그는 *laravel-cli-2017-08-03.log* 와 같이 별도의 파일로 관리되므로 권한 문제가 발생하지 않습니다.

참고

- [라라벨 5 eloquent orm query log 남기기](#)
- [로깅\(Logging\) 남기기](#)
- [setfacl 과 getfacl](#)
- <https://stackoverflow.com/questions/27674597/laravel-daily-log-created-with-wrong-permissions>