

Journalisation (logging) dans Laravel:

1.Introduction :

Le journal de Laravel 9 fait partie intégrante de l'application Laravel. Fondamentalement, il est utilisé comme une application pour surveiller l'activité de l'application. Laravel dispose de services de log robustes pour convertir les messages de log dans un fichier. Aujourd'hui, nous allons faire la démonstration de la journalisation de Laravel. Ce cours vous aidera à comprendre l'état de votre application. Ce qui se passe avec votre application. S'il y avait une erreur dans votre logiciel, vous verriez le message d'erreur du système dans votre fichier journal.

Comment cela fonctionne-t-il ?

Laravel 9 permet aux développeurs d'enregistrer des messages sur le comportement du système de n'importe quelle application dans des fichiers, comme le système de journaux d'erreurs, et de les envoyer pour informer les équipes de développement.

Il est basé sur les canaux de Laravel 9. Les canaux sont une façon spécifique d'écrire le message de journal d'un système. Chaque canal représente une destination différente, et nous pouvons envoyer des messages à différents canaux simultanément.

2.Écriture de messages de journal

Vous pouvez écrire des informations dans le journal à l'aide de la façade **Log**. Comme mentionné précédemment, le journal fournit les huit niveaux de

journalisation définis dans la spécification RFC 5424 : emergency, alert, critical, error, warning, notice, info et debug :

```
use Illuminate\Support\Facades\Log;

Log::emergency($message);
Log::alert($message);
Log::critical($message);
Log::error($message);
Log::warning($message);
Log::notice($message);
Log::info($message);
Log::debug($message);
```

Vous pouvez appeler n'importe laquelle de ces méthodes pour enregistrer un message pour le niveau correspondant. Par défaut, le message sera écrit dans le canal de journalisation par défaut tel que configuré par votre fichier de configuration de journalisation :

```
public function show(Request $request,$student)
{
    try {
        $student=Student::findOrFail($student);
        Log::info('User see his profile :'.$student->id);

    } catch (ModelNotFoundException $e) {

        throw new studentNotFoundException(100);
    }catch (\Exception $e) {

        dd(get_class($e));
    }
    return view('students.profile', ['student' => $student]);
}
```

//Output (laravel.log)

```
[2023-02-11 20:53:44] local.INFO: User see his profile :74
```

3. Informations contextuelles

Un tableau de données contextuelles peut être transmis aux méthodes de journalisation. Ces données contextuelles seront formatées et affichées avec le message du journal :

```
use Illuminate\Support\Facades\Log;

Log::info('User see his profile.', ['id' => $student->id, 'ip' => $request->ip()]);
```

`\storage\logs\laravel.log`

```
storage > logs > ≡ laravel.log
1  [2023-03-20 14:57:11] local.INFO: User see his profile :adil1
2  [2023-03-20 14:58:38] local.WARNING: User see his profile :adil1
3  [2023-03-20 14:59:38] local.WARNING: User see his profile :Fatima
4  [2023-03-20 15:04:23] local.INFO: User see his profile. {"id":92,"ip":"127.0.0.1"}
5  [2023-03-20 15:21:40] local.CRITICAL: User see his profile. {"id":74,"ip":"127.0.0.1"}
6
```

Exercice :

Le Log Viewer d'OPcodes est un compagnon parfait pour votre application Laravel.

Vous n'aurez plus besoin de lire les fichiers logs bruts de Laravel en essayant de trouver ce que vous cherchez.

Log Viewer vous aide à voir rapidement et clairement les entrées de log individuelles, à rechercher, filtrer et donner un sens à vos logs Laravel rapidement. Il est gratuit et facile à installer.

Installation :

Installer le package via composer:

```
composer require opcodesio/log-viewer
```

Une fois l'installation terminée, vous pourrez accéder à Log Viewer directement dans votre navigateur.

Par défaut, l'application est disponible à l'adresse suivante :

`{APP_URL}/log-viewer`.

Better Log Viewer

← Back to Laravel

laravel.log	2.58 MB	⌵
laravel-2022-07-28.log	104.80 MB	⌵
laravel-2022-07-26.log	50.40 MB	⌵
laravel-2022-07-24.log	1,009.20 MB	⌵
laravel-2022-07-11.log	31.76 MB	⌵
laravel-2022-07-09.log	72.09 MB	⌵
laravel-2022-03-06.log	42.57 KB	⌵
laravel-2022-03-05.log	42.57 KB	⌵
laravel-2022-03-02.log	42.57 KB	⌵
laravel-2022-03-01.log	295.74 KB	⌵
laravel-2022-02-05.log	38.59 KB	⌵
laravel-2022-01-28.log	129.88 KB	⌵
laravel-2022-01-27.log	79.21 KB	⌵
laravel-2022-01-23.log	93.63 KB	⌵
laravel-2022-01-19.log	45.88 KB	⌵
laravel-2022-01-16.log	368.74 KB	⌵
laravel-2022-01-15.log	234.77 KB	⌵
laravel-2022-01-11.log	454.05 KB	⌵

Debug: 333,672

Info: 91,756

Warning: 60,858

Error: 672

Search... RegEx welcome!

Level	Time	Env	Description	Newest first	50 items per page
Error	2022-07-09 23:57:02	production	SQLSTATE[HY093]: Invalid parameter number (SQL: ANALYZE TABLE 'Action toplist')	487,156	
Debug	2022-07-09 23:48:48	production	Handling facebook login.	487,155	
Error	2022-07-09 23:47:48	production	Maximum execution time of 100 seconds exceeded	487,154	
Debug	2022-07-09 23:47:46	production	Initiating facebook login.	487,153	
Debug	2022-07-09 23:46:11	production	[test@example.com] credentials_not_correct	487,152	Copy link to this log entry
[test@example.com] credentials_not_correct [HTTP request!] +User ID: guest +Request: POST http://127.0.0.1/login +Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36					
Warning	2022-07-09 23:46:03	production	Exception: User not found	487,151	
Warning	2022-07-09 23:41:58	production	Exception: User not found	487,150	
Warning	2022-07-09 23:41:33	production	Exception: User not found	487,149	
Debug	2022-07-09 23:37:54	production	Two-Factor Auth request.	487,148	
Warning	2022-07-09 23:37:06	production	Exception: User not found	487,147	
Warning	2022-07-09 23:36:57	production	Exception: User not found	487,146	
Debug	2022-07-09 23:32:46	production	Two-Factor Auth request.	487,145	
Debug	2022-07-09 23:32:43	production	Two-Factor Auth request.	487,144	
Debug	2022-07-09 23:31:22	production	[UID 50314] - Got 0 clients after filters have been applied	487,143	
Debug	2022-07-09 23:31:22	production	[UID 50314] - Got 0 clients who should be deleted today	487,142	
Debug	2022-07-09 23:31:22	production	[UID 50314] Starting deletion of marked clients.	487,141	
Debug	2022-07-09 23:31:22	production	[UID 50038] - Got 0 clients after filters have been applied	487,140	
Debug	2022-07-09 23:31:22	production	[UID 50038] - Got 0 clients who should be deleted today	487,139	
Debug	2022-07-09 23:31:22	production	[UID 49600] - Got 0 clients after filters have been applied	487,138	

1 2 3 4 5 6 7 8 9 10 ... 7,908 7,909 →

Memory: 105.52 MB, Duration: 209ms