

# ✨ Manual de Uso - API Genérica de Log para Laravel

## 01 Visão Geral

Esta API de log foi desenvolvida para registrar, auditar e consultar todas as requisições feitas por usuários autenticados em aplicações Laravel. É compatível com qualquer sistema Laravel 8+ e possui configuração via arquivos.

Ela captura:

- IP do usuário
  - Caminho e método HTTP
  - Status HTTP
  - Conteúdo da requisição e da resposta
  - Estado anterior do recurso (antes de alterações)
-

## Estrutura do Módulo: **modules/LogModule/**

modules/

```
└─ LogModule/
  │   └─ Controllers/
  │       └─ LogController.php
  │   └─ Middleware/
  │       └─ AccessLog.php
  │   └─ Models/
  │       └─ AccessLog.php
  │   └─ Services/
  │       └─ AccessLogService.php
  │   └─ Config/
  │       │   └─ log_access_filter.php
  │       │   └─ log_http_verbs.php
  │       │   └─ log_models_mapping.php
  │       │   └─ log_routes_mapping.php
  │       │   └─ log_status.php
  │   └─ database/
  │       │   └─ migrations/
  │       │       └─ 2025_01_01_000000_create_access_logs_table.php
  │   └─ routes/
  │       └─ api.php
  └─ README.md
```

## Instalação

### 1. Copie os arquivos

- `app/Http/Middleware/AccessLog.php`
- `app/Http/Controllers/Api/LogController.php`
- `app/Services/AccessLogService.php`
- `app/Models/AccessLog.php`
- `database/migrations/create_access_logs_table.php`
- Arquivos da pasta `config/` iniciados por `log_`

### 2. Adicione o Middleware no `Kernel.php`

```
protected $middlewareGroups = [  
    'api' => [  
        \App\Http\Middleware\AccessLog::class,  
        // outros middlewares...  
    ],  
];
```

### 3. Rode a migration

```
php artisan migrate
```

---

## Configuração (Arquivos **config/**)

### **log\_access\_filter.php**

- Define perfis autorizados, rotas com permissão e rotas ocultas dos logs visuais.

### **log\_http\_verbs.php**

- Mapeia GET, POST, PUT, DELETE etc. para mensagens descritivas.

### **log\_status.php**

- Traduz códigos HTTP para mensagens legíveis e sucesso ou falha.

### **log\_models\_mapping.php**

- Mapeia rotas para os respectivos **models** Eloquent, utilizados para buscar o estado anterior.

### **log\_routes\_mapping.php**

- Define descrições legíveis para cada rota + verbo HTTP, usadas na exibição dos logs.

---

## Exemplo de Registro

Ao realizar:

PUT /api/users/42

Será registrado:

- IP do usuário
- Verbo HTTP: PUT
- Caminho: api/users/42
- Status da resposta: 200

- Dados enviados (request)
  - Dados retornados (response)
  - Dados anteriores do registro (User::find(42))
- 

## Endpoints

### GET /api/logs/formatted

Retorna os 50 registros mais recentes, já formatados:

```
[
  {
    "message": "O(A) Ana do(a) Unidade A atualizou um perfil com sucesso.",
    "dateTime": "14/04/2025 às 10h33"
  }
]
```

### GET /api/logs/search

Aceita filtros:

```
{
  "user_name": "ana",
  "start_date": "2025-01-01",
  "end_date": "2025-02-01",
  "method": "POST",
  "status": "200",
  "path": "api/users"
}
```

---

## Como Adicionar uma Nova Rota para Auditoria

1. Registre a rota em `log_models_mapping.php`:

```
'api/hospitals/{id}' => [  
    'model' => App\Models\Hospital::class,  
    'table' => 'hospitals',  
    'primary_key' => 'id',  
],
```

2. Adicione uma descrição em `log_routes_mapping.php`:

```
'api/hospitals/{id}' => [  
    'PUT' => 'um hospital atualizado',  
    'DELETE' => 'um hospital excluído',  
],
```

---

## Requisitos

- Laravel 8+
  - PHP 7.4 ou superior
  - Middleware `AccessLog` adicionado ao grupo `api`
  - Migration da tabela `access_logs` executada com sucesso
- 

## Licença

Secretária de Saúde do Estado da Bahia - SESAB/CGTICS/DMA/COSIS