

Comandos para Windows - Verificar Laravel

1. Verificar Rotas dos Treinos (Windows)

Opção 1 - CMD (Prompt de Comando):

Plain Text

```
php artisan route:list | findstr treino
```

Opção 2 - PowerShell:

Plain Text

```
php artisan route:list | Select-String "treino"
```

Opção 3 - Ver todas as rotas e procurar manualmente:

Plain Text

```
php artisan route:list
```

(Depois procure visualmente por "treino" na lista)

2. Verificar se Existe Controller

Verificar se arquivo existe:

Plain Text

```
dir app\Http\Controllers\TreinoController.php
```

Ou listar todos os controllers:

Plain Text

```
dir app\Http\Controllers\*.php
```

3. 🗄️ Verificar Model

Verificar se existe:

Plain Text

```
dir app\Models\Treino.php
```

Listar todos os models:

Plain Text

```
dir app\Models\*.php
```

4. 📊 Ver Logs do Laravel (Windows)

Ver últimas linhas do log:

Plain Text

```
type storage\logs\laravel.log | more
```

Ver logs em tempo real (PowerShell):

Plain Text

```
Get-Content storage\logs\laravel.log -Wait -Tail 10
```

5. 🧪 Testar Rotas Manualmente

Usando curl (se instalado):

Plain Text

```
curl http://10.125.135.38:8000/api/status
```

Usando PowerShell (sem curl):

Plain Text

```
Invoke-RestMethod -Uri "http://10.125.135.38:8000/api/status" -Method Get
```

Testar treinos (com token):

Plain Text

```
$headers = @{  
    "Authorization" = "Bearer SEU_TOKEN_AQUI"  
    "Accept" = "application/json"  
}  
Invoke-RestMethod -Uri "http://10.125.135.38:8000/api/treinos" -Method Get -  
Headers $headers
```

6. Comandos de Limpeza (Windows)

Plain Text

```
php artisan cache:clear  
php artisan config:clear  
php artisan route:clear  
php artisan view:clear
```

7. Teste Mais Simples

Abra o browser e acesse diretamente:

Plain Text

```
http://10.125.135.38:8000/api/status
http://10.125.135.38:8000/api/treinios
```

8. Criar Arquivo de Teste

Crie um arquivo `test-routes.php` na raiz do projeto:

PHP

```
<?php
// test-routes.php

require_once 'vendor/autoload.php';

$app = require_once 'bootstrap/app.php';
$kernel = $app->make(Illuminate\Contracts\Console\Kernel::class);
$kernel->bootstrap();

// Listar todas as rotas
$routes = Route::getRoutes();

echo "=== ROTAS ENCONTRADAS ===\n";
foreach ($routes as $route) {
    $uri = $route->uri();
    $methods = implode('|', $route->methods());

    if (strpos($uri, 'treino') !== false) {
        echo "[$methods] $uri\n";
    }
}

echo "\n=== TODAS AS ROTAS API ===\n";
foreach ($routes as $route) {
    $uri = $route->uri();
    $methods = implode('|', $route->methods());

    if (strpos($uri, 'api/') === 0) {
        echo "[$methods] $uri\n";
    }
}
?>
```

Execute:

Plain Text

```
php test-routes.php
```

9. ⚡ Verificação Rápida - Estrutura de Arquivos

Verificar se existem os arquivos principais:

Plain Text

```
echo === CONTROLLERS ===  
dir app\Http\Controllers\*Treino*.php  
  
echo === MODELS ===  
dir app\Models\*Treino*.php  
  
echo === MIGRATIONS ===  
dir database\Migrations\*treino*.php  
  
echo === ROUTES ===  
type routes/api.php | findstr treino
```

10. 🔍 Alternativa - Usar Notepad++

Se tiver o Notepad++ instalado:

1. Abra o arquivo `routes/api.php`
2. Use Ctrl+F para procurar por "treino"
3. Veja se existem rotas definidas

🎯 Comandos Essenciais para Executar AGORA:

Plain Text

```
# 1. Ver todas as rotas
php artisan route:list

# 2. Procurar por treino nas rotas
php artisan route:list | findstr treino

# 3. Verificar se controller existe
dir app\Http\Controllers\TreinoController.php

# 4. Ver conteúdo do arquivo de rotas
type routes\api.php
```

Execute esses comandos e me diga o que aparece!