

---

# Créer un webhook GitHub avec Symfony

Guide pour créer une route webhook GitHub dans Symfony pour automatiser les déploiements

**Symfony** **Programmation** **15 min de lecture** **Niveau Intermédiaire**

---

Document généré le 13/05/2026 à 11h11 · [novy.fr/wiki/webhook-github-symfony](https://novy.fr/wiki/webhook-github-symfony)

# Sommaire

27 section(s) · 15 min de lecture

## ⚠ Prérequis de sécurité

## 📄 Étapes d'implémentation

- ↳ 1. Créer le contrôleur Symfony
- ↳ 2. Implémenter la logique du webhook
- ↳ 3. Configurer le secret dans le fichier .env

## 📄 Configuration GitHub

- ↳ Créer le webhook dans GitHub

## 📄 Structure du projet

- ↳ Arborescence typique d'un projet Symfony
- ↳ Configuration du chemin du projet
- ↳ Vérifier le bon chemin

## 📄 Gestion des droits d'accès

- ↳ Droits nécessaires
- ↳ Tester les droits manuellement
- ↳ Configurer les droits si nécessaire

## 📄 Test du webhook

- ↳ Test manuel avec curl
- ↳ Vérifier les logs

## 📄 Sécurité avancée

- ↳ Bonnes pratiques
- ↳ Validation supplémentaire

## 📄 Déploiement automatique

- ↳ Script de déploiement complet

## 📄 Dépannage

- ↳ Problèmes courants
- ↳ Commandes de débogage

## 📄 Conclusion

Ce guide vous explique comment créer une route webhook GitHub dans un projet Symfony pour exécuter automatiquement un `git pull` lorsque GitHub envoie une requête (par exemple après un push sur un dépôt).

## ⚠ Prérequis de sécurité

---

Avant de commencer, assurez-vous de :

- **Protéger la route** avec un secret partagé (le webhook secret de GitHub)
- **Limiter l'exécution** à un répertoire spécifique
- **Donner les droits d'écriture** uniquement à l'utilisateur approprié
- **Utiliser HTTPS** sur votre domaine
- **Utiliser un utilisateur système** avec des droits limités

## 📄 Étapes d'implémentation

---

### 1. Créer le contrôleur Symfony

Générez le contrôleur avec la commande Symfony :

```
php bin/console make:controller GitWebhookController
```

📄 Copier

Cela va créer le fichier `src/Controller/GitWebhookController.php`.

### 2. Implémenter la logique du webhook

Remplacez le contenu du contrôleur par le code suivant :

```

<?php

namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

class GitWebhookController extends AbstractController
{
    #[Route('/webhook/github', name: 'github_webhook', methods: ['POST'])]
    public function githubWebhook(Request $request): Response
    {
        // Récupérer le secret depuis les variables d'environnement
        $secret = $_ENV['GITHUB_WEBHOOK_SECRET'];

        // Récupérer le payload et la signature
        $payload = $request->getContent();
        $signature = $request->headers->get('X-Hub-Signature-256');

        // Vérifier la signature pour sécuriser le webhook
        if (!$signature || !hash_equals('sha256=' . hash_hmac('sha256', $payload, $secret),
        $signature)) {
            return new Response('Invalid signature', 403);
        }

        // Chemin vers le répertoire de votre projet
        $repoPath = '/var/www/html/mon-projet-symfony'; // Ajustez selon votre structure

        // Exécuter git pull
        $output = [];
        $resultCode = 0;

        exec("cd $repoPath && git pull 2>&1", $output, $resultCode);

        // Vérifier le résultat de l'exécution
        if ($resultCode !== 0) {
            return new Response("Git pull failed:\n" . implode("\n", $output), 500);
        }

        return new Response("Git pull successful:\n" . implode("\n", $output), 200);
    }
}

```

📋 Copier

### 3. Configurer le secret dans le fichier .env

Ajoutez la variable d'environnement dans votre fichier `.env` :

```
GITHUB_WEBHOOK_SECRET=your_webhook_secret_here
```

📋 Copier

**Important** : Remplacez `your_webhook_secret_here` par un secret fort et unique.

## □ Configuration GitHub

---

### Créer le webhook dans GitHub

1. Allez dans votre dépôt GitHub
2. Naviguez vers **Settings > Webhooks**
3. Cliquez sur **Add webhook**
4. Configurez les paramètres suivants :

#### Paramètres de base

- **Payload URL** : `https://votre-domaine.com/webhook/github`
- **Content type** : `application/json`
- **Secret** : Le même secret que dans votre fichier `.env`
- **SSL verification** : Enable SSL verification

#### Événements

- **Events** : Choisissez selon vos besoins
  - **Just the push event** : Pour déclencher uniquement sur les push
  - **Send me everything** : Pour tous les événements
  - **Let me select individual events** : Pour sélectionner des événements spécifiques

## □ Structure du projet

---

### Arborescence typique d'un projet Symfony

```
/var/www/html/mon-projet-symfony/  
├─ bin/  
├─ config/  
├─ public/  
├─ src/  
│   └─ Controller/  
│       └─ GitWebhookController.php  
├─ templates/  
├─ var/  
├─ vendor/  
└─ .git/
```

📄 Copier

### Configuration du chemin du projet

Le chemin `$repoPath` doit pointer vers la racine du dépôt Git (où se trouve le dossier `.git`).

#### Exemple :

```
$repoPath = '/var/www/html/mon-projet-symfony';
```

📄 Copier

## Vérifier le bon chemin

Connectez-vous à votre serveur et vérifiez :

```
cd /var/www/html/mon-projet-symfony  
ls -a
```

📄 Copier

Vous devez voir le dossier `.git` dans la liste.

## 📄 Gestion des droits d'accès

---

### Droits nécessaires

L'utilisateur qui exécute le processus PHP (souvent `www-data`) doit avoir :

- **Accès en lecture/écriture** au dossier du projet
- **Clé SSH installée** si vous clonez en SSH (`git@github.com:...`)

### Tester les droits manuellement

Testez les droits avec la commande suivante :

```
sudo -u www-data bash -c 'cd /var/www/html/mon-projet-symfony && git pull'
```

📄 Copier

### Configurer les droits si nécessaire

Si vous avez des problèmes de droits :

```
# Donner les droits à www-data  
sudo chown -R www-data:www-data /var/www/html/mon-projet-symfony  
  
# Ou ajouter www-data au groupe approprié  
sudo usermod -a -G git www-data
```

📄 Copier

## 📄 Test du webhook

---

### Test manuel avec curl

```
curl -X POST \  
  -H "Content-Type: application/json" \  
  -H "X-Hub-Signature-256: sha256=$(echo -n 'test' | openssl dgst -sha256 -hmac  
'your_secret' | cut -d' ' -f2)" \  
  -d '{"test": "data"}' \  
  https://votre-domaine.com/webhook/github
```

📄 Copier

## Vérifier les logs

Consultez les logs Symfony pour déboguer :

```
tail -f var/log/dev.log  
# ou  
tail -f var/log/prod.log
```

📄 Copier

## ☐ Sécurité avancée

---

### Bonnes pratiques

1. **Utilisez HTTPS** obligatoirement
2. **Changez régulièrement** le secret du webhook
3. **Limitez les événements** déclencheurs
4. **Surveillez les logs** d'accès
5. **Utilisez un firewall** pour limiter l'accès

### Validation supplémentaire

Vous pouvez ajouter des validations supplémentaires :

```
// Vérifier le type d'événement  
$event = $request->headers->get('X-GitHub-Event');  
if ($event !== 'push') {  
    return new Response('Event not supported', 400);  
}  
  
// Vérifier la branche  
$payloadData = json_decode($payload, true);  
$branch = $payloadData['ref'] ?? '';  
if ($branch !== 'refs/heads/main') {  
    return new Response('Branch not supported', 400);  
}
```

📄 Copier

## ☐ Déploiement automatique

---

## Script de déploiement complet

Pour un déploiement plus robuste, vous pouvez créer un script séparé :

```
// Dans le contrôleur
exec("cd $repoPath && ./deploy.sh 2>&1", $output, $resultCode);
```

📄 Copier

```
#!/bin/bash
# deploy.sh
git pull
composer install --no-dev --optimize-autoloader
php bin/console cache:clear --env=prod
php bin/console doctrine:migrations:migrate --no-interaction
```

📄 Copier

## 📄 Dépannage

---

### Problèmes courants

#### 1. Erreur 403 - Invalid signature

- Vérifiez que le secret correspond entre GitHub et votre `.env`
- Assurez-vous que la signature est correctement calculée

#### 2. Erreur 500 - Git pull failed

- Vérifiez les droits d'accès au dossier
- Testez manuellement la commande `git pull`
- Consultez les logs d'erreur

#### 3. Webhook non déclenché

- Vérifiez l'URL du webhook
- Assurez-vous que HTTPS est activé
- Vérifiez les événements configurés

### Commandes de débogage

```
# Vérifier les droits
ls -la /var/www/html/mon-projet-symfony

# Tester git pull manuellement
sudo -u www-data git pull

# Vérifier les logs PHP
tail -f /var/log/apache2/error.log
```

📄 Copier

## □ Conclusion

---

Ce webhook GitHub vous permet d'automatiser le déploiement de votre application Symfony. Assurez-vous de toujours maintenir un niveau de sécurité élevé et de tester régulièrement le bon fonctionnement du système.