

## Démonstration complète avec Apache : redirection, réécriture, gestion des pages d'erreur, conditions et alias

Voici un projet complet intégrant toutes les fonctionnalités que vous avez demandées dans un contexte pratique.

---

### Objectif du projet

#### 1. Redirection :

- Rediriger toutes les requêtes HTTP vers HTTPS.
- Rediriger un domaine entier vers un autre domaine.

#### 2. Réécriture :

- Réécrire des URLs pour des URLs propres.
- Réécrire en fonction de la langue ou d'autres paramètres.

#### 3. Gestion des erreurs :

- Personnaliser les pages d'erreur (404, 403, 500).

#### 4. Conditions :

- Ajouter des règles conditionnelles basées sur l'IP, l'heure ou l'agent utilisateur.

#### 5. Alias :

- Servir un fichier statique ou un autre répertoire via un chemin personnalisé.
- 

### Étape 1 : Structure du projet

#### 1. Arborescence des fichiers :

```
/var/www/html/my-project/  
├── index.php  
├── about.php  
├── fr/  
│   └── contact.php  
├── en/  
│   └── contact.php  
├── static/  
│   └── info.html  
└── errors/  
    ├── 404.html  
    ├── 403.html  
    └── 500.html
```

#### 2. Créer les fichiers nécessaires :

- Créez le répertoire et les fichiers correspondants :

```
mkdir -p /var/www/html/my-project/{fr,en,static,errors}
echo "<?php echo 'Home Page'; ?>" > /var/www/html/my-
project/index.php
echo "<?php echo 'About Page'; ?>" > /var/www/html/my-
project/about.php
echo "<?php echo 'Contact Page - FR'; ?>" > /var/www/html/my-
project/fr/contact.php
echo "<?php echo 'Contact Page - EN'; ?>" > /var/www/html/my-
project/en/contact.php
echo "<h1>Static Info</h1>" > /var/www/html/my-
project/static/info.html
echo "<h1>Page Not Found</h1>" > /var/www/html/my-
project/errors/404.html
echo "<h1>Forbidden</h1>" > /var/www/html/my-
project/errors/403.html
echo "<h1>Server Error</h1>" > /var/www/html/my-
project/errors/500.html
```

---

## Étape 2 : Configuration Apache

1. **Créer un fichier de configuration pour le site** : Créez le fichier `/etc/apache2/sites-available/my-project.conf` :

```
<VirtualHost *:80>
    ServerName my-project.local
    DocumentRoot /var/www/html/my-project

    # Redirection HTTP vers HTTPS
    RewriteEngine On
    RewriteCond %{HTTPS} !=on
    RewriteRule ^(.*)$ https://%{HTTP_HOST}/$1 [R=301,L]
</VirtualHost>

<VirtualHost *:443>
    ServerName my-project.local
    DocumentRoot /var/www/html/my-project

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/my-project.pem
    SSLCertificateKeyFile /etc/ssl/private/my-project.key

    # Réécriture des URLs
    RewriteEngine On

    # URLs propres pour about.php
    RewriteRule ^about$ /about.php [L]
```

```
# Multilingue pour contact (FR et EN)
RewriteRule ^contact/fr$ /fr/contact.php [L]
RewriteRule ^contact/en$ /en/contact.php [L]

# Gestion conditionnelle : Bloquer une IP
RewriteCond %{REMOTE_ADDR} ^192\.168\.1\.100$
RewriteRule ^(.*)$ /errors/403.html [R=403,L]

# Gestion conditionnelle : Redirection mobile
RewriteCond %{HTTP_USER_AGENT} "Mobile" [NC]
RewriteRule ^(.*)$ /mobile-version.html [L]

# Gestion des pages d'erreur personnalisées
ErrorDocument 404 /errors/404.html
ErrorDocument 403 /errors/403.html
ErrorDocument 500 /errors/500.html

# Alias pour un fichier statique
Alias /info /var/www/html/my-project/static/info.html

<Directory /var/www/html/my-project>
    AllowOverride All
    Require all granted
</Directory>
</VirtualHost>
```

## 2. Activer le site et recharger Apache :

```
sudo a2ensite my-project
sudo systemctl reload apache2
```

---

## Étape 3 : Fonctionnalités couvertes

### 1. Redirection HTTP vers HTTPS

- Lorsqu'un utilisateur accède à <http://my-project.local>, il est automatiquement redirigé vers <https://my-project.local>.

---

### 2. Réécriture d'URL

- **URLs propres :**
    - Accéder à <https://my-project.local/about> charge [/about.php](#).
  - **Multilingue :**
    - Accéder à <https://my-project.local/contact/fr> charge [/fr/contact.php](#).
    - Accéder à <https://my-project.local/contact/en> charge [/en/contact.php](#).
-

### 3. Gestion conditionnelle

- **Bloquer une IP :**
    - Les utilisateurs ayant l'adresse IP `192.168.1.100` verront une page d'erreur `403 Forbidden`.
  - **Redirection mobile :**
    - Les utilisateurs mobiles sont redirigés vers `/mobile-version.html`.
- 

### 4. Gestion des pages d'erreur

- Une erreur `404` affiche la page personnalisée `/errors/404.html`.
  - Une erreur `403` affiche la page personnalisée `/errors/403.html`.
  - Une erreur `500` affiche la page personnalisée `/errors/500.html`.
- 

### 5. Alias

- Accéder à `https://my-project.local/info` charge le fichier statique `/static/info.html`.
- 

## Étape 4 : Tests

#### 1. Vérifiez la redirection HTTP → HTTPS :

- Accédez à `http://my-project.local` et assurez-vous d'être redirigé vers `https://my-project.local`.

#### 2. Testez les URLs réécrites :

- Accédez à :
  - `https://my-project.local/about`
  - `https://my-project.local/contact/fr`
  - `https://my-project.local/contact/en`

#### 3. Testez les conditions :

- Simulez une IP bloquée (`192.168.1.100`) :
  - Modifiez votre adresse IP locale ou utilisez `curl` :

```
curl -H "X-Forwarded-For: 192.168.1.100" https://my-project.local
```

- Simulez un utilisateur mobile :

```
curl -A "Mobile" https://my-project.local
```

#### 4. Testez les pages d'erreur :

- Accédez à une URL inexistante (<https://my-project.local/nonexistent>).

## 5. Testez l'alias :

- Accédez à <https://my-project.local/info>.

---

## Résumé

Cette configuration couvre :

- **Redirection** : HTTP → HTTPS.
- **Réécriture** : URLs propres et multilingues.
- **Conditions** : Blocage IP et redirection mobile.
- **Pages d'erreur** : Pages personnalisées pour 404, 403, 500.
- **Alias** : Accès rapide à un fichier statique.