NGINX — Rediriger les 404 du reverse proxy vers /index.php

Objectif : quand l'application derrière le reverse proxy renvoie une erreur 404, Nginx intercepte cette 404 et redirige le client vers /index.php (même domaine). Cette méthode est idéale pour les sites où l'index gère le routage (frameworks, CMS, etc.).

> Important : activez « proxy_intercept_errors on; » pour que Nginx capte les 404 provenant du backend.

Contexte & Problème

Dans une architecture reverse proxy, certaines URLs inexistantes côté application renvoient 404. Sans configuration spécifique, Nginx relaie simplement l'erreur au client. Nous souhaitons au contraire RENVOYER le client vers /index.php pour laisser l'application gérer la route.

Solution mise en place (Redirection 302)

Configuration côté serveur virtuel :

```
server {
    server_name exemple.tld;

    location / {
        proxy_pass http://backend;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_intercept_errors on;  # ← indispensable pour capter les
404 du backend
    }

    # Redirige les 404 vers /index.php (302 visible côté client)
    error_page 404 =302 $scheme://$host/index.php;
}
```


```
server {
    server_name exemple.tld;

    location / {
        proxy_pass http://backend;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_intercept_errors on;  # ← indispensable pour capter les
```

```
404 du backend
    }

    # Redirige les 404 vers /index.php (302 visible côté client)
    error_page 404 =302 $scheme://$host/index.php;
}
```

Étapes d'application

1. Ouvrir le fichier de configuration du serveur (ex. /etc/nginx/sites-available/mon-site).

2. Dans le bloc server { }, vérifier que le proxy vers votre backend est correct (proxy_pass, en-têtes).

3. Activer proxy_intercept_errors on; dans le location / ou dans le bloc server.

4. Ajouter la directive error_page 404 =302 $scheme://$host/index.php;

5. Valider la syntaxe puis recharger Nginx.

```
nginx -t && systemctl reload nginx
```


```
nginx -t && systemctl reload nginx
```


Variante : router en interne (sans 302)

Si vous préférez ne PAS exposer de redirection au client, vous pouvez router en interne vers index.php sur le backend :

```
location @fallback_index {
    proxy_pass http://backend/index.php?$is_args$args;
}

error_page 404 = @fallback_index;
```


```
location @fallback_index {
    proxy_pass http://backend/index.php?$is_args$args;
}

error_page 404 = @fallback_index;
```


Variante : URL précise

Pour forcer vers une URL externe spécifique :

```
error_page 404 =302 https://www.tvpronet.com/index.php;
```


```
error_page 404 =302 https://www.tvpronet.com/index.php;
```

Tests rapides

1) URL inexistante :

```
curl -I https://exemple.tld/chemin/inexistant
```

```
curl -I https://exemple.tld/chemin/inexistant
```

2) Vérifier le header Location (pour la version redirection 302).

3) Tester depuis le navigateur en navigation privée.

Bonnes pratiques & remarques

• Conserver les paramètres de requête si besoin (ex. /index.php?is_argsargs).

• Limiter aux seules 404 (ne pas attraper 500/502 sauf besoin).

• Ne pas boucler : assurez-vous que /index.php existe et renvoie 200.

• Tracer côté logs pour vérifier le comportement (access.log et error.log).

Rollback

Supprimez la directive error_page et désactivez proxy_intercept_errors si vous revenez au comportement par défaut, puis reload.