

Routes

Réécriture d'URL

Au lieu de

`http://mon-site.com/produits.php`

Un URL plus propre

`http://mon-site.com/produits`

Avantages

- Découpler les url du système de fichier
- Ne pas exposer le langage utilisé pour développer l'application
- Des URL plus robots friendly

Mise en place avec Apache

Vérifier que mod_rewrite est activé

Dans le fichier httpd.conf

```
LoadModule rewrite_module modules/mod_rewrite.so
```

Réécriture simple

Créer un fichier .htaccess dans le dossier racine du serveur (htdocs par défaut)

```
# Activation du mod rewrite d'Apache
RewriteEngine On`
```

```
# Règle de réécriture
```

```
# Si l'expression régulière en premier argument est vérifiée
```

```
# on redirige vers la page en deuxième argument
```

```
RewriteRule ^/produits$ /produits.php [L]
```

Réécriture avec passage de paramètre

```
RewriteEngine On
```

```
# l'url /article-edit-5
```

```
# pointera vers
```

```
# /article-edit.php?id=5
```

```
RewriteRule '^/article-edit-(\d+)$ /article-edit.php?id=$1 [L]
```

Un autre exemple

```
RewriteEngine On
```

```
# l'url /article-list-page5
```

```
# pointera vers
```

```
# /article-edit.php?page=5
```

```
RewriteRule '^/article-list-page(\d+)$ /articles.php?page=$1 [L]
```

Avec deux paramètres

```
RewriteEngine On
```

```
# l'url /article-list-php-page5
```

```
# pointera vers
```

```
# /article-edit.php?category=php&page=5
```

```
RewriteRule '^/article-list-([a-z])page(\d+)$ /articles.php?category=$1 &page=$2 [L]
```


Les flags

- L : indique que la règle est finale aucune autre règle ne sera testée
- R : envoie un code de redirection ex: [R=301, L]
- F : forbidden envoie un code d'erreur 403, il est alors inutile de préciser une redirection
- G : gone envoie un code d'erreur 410
- QSA : querystring appending ajoute l'éventuelle chaîne de requête à l'url réécrite

Les conditions de réécriture

Les conditions s'appliquent à la prochaine règle de réécriture

N'applique la réécriture que si l'adresse est locale

```
RewriteCond %{REMOTE_ADDR} localhost
```

```
RewriteRule ^(.*)$ [F]
```

N'applique la réécriture que si l'adresse est n'est pas locale

```
RewriteCond %{REMOTE_ADDR} !localhost
```

```
RewriteRule ^(.*)$ [F]
```

Autre exemple

```
# N'applique la règle que si le resource  
# n'est ni un fichier ni un dossier existant  
RewriteCond %{REQUEST_FILENAME} !-d  
RewriteCond %{REQUEST_FILENAME} !-f  
RewriteRule ^(.*)$ index.php?path=$1 [L,QSA]
```

Contrôleur frontal

```
RewriteEngine On
```

```
RewriteCond %{REQUEST_FILENAME} !-d
```

```
RewriteCond %{REQUEST_FILENAME} !-f
```

```
RewriteRule ^(.*)$ app.php?controller=$1 [L,QSA]
```

Contrôleur frontal php

```
//app.php
```

```
//récupération du nom du contrôleur
```

```
$controllerName = filter_input(INPUT_GET, 'controller');
```

```
//récupération du dossier parent
```

```
define('ROOT_PATH', dirname(__DIR__));
```

```
$controllerPath = ROOT_PATH. "/src/${controllerName}.php";
```

```
if(file_exists($controllerPath)){  
    require $controllerPath;  
}
```

Avantages du contrôleur frontal

- Le code source de l'application n'est pas exposé sur le web
- Les opérations récurrentes sont centralisées :
 - authentification
 - gestion des dépendances
 - configuration

Hôte virtuel

Servir des pages en dehors de htdocs

Dans le fichier `conf/extra/httpd-vhosts.conf`

```
<VirtualHost *:80>
    DocumentRoot "/projets/app/web"
    ServerName projet.local

    <Directory "/projets/app/web">
        Options All
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Fichier hosts

- /etc/hosts pour linux et Mac OS
- C:\Windows\System 32\drivers\etc\hosts pour Windows

```
127.0.0.1    projet.local
```