

Le passage de paramètres dans les pages statiques

Post original : <http://forum.pluxml.org/viewtopic.php?pid=27105>

Deux problèmes sont évoqués dans ce post. Le problème évoqué directement est celui de la non-validité de & dans un lien en XHTML. Le second problème, non évoqué directement dans ce post, est celui de l'obligation de passage de paramètres par un & et non pas par un ? comme il se doit.

I. Bug lors du passage de paramètres

Plaçons un code très simple dans une page statique après avoir activé l'URL rewriting.

```
<?php echo $_GET['dep'] ;?>
```

La logique voudrait que lorsque l'URL <http://localhost/static1/statique-1?dep=26> est appelée, la page statique affiche '26'. Pourtant la page affiche `Undefined index: dep`.

Maintenant, essayons l'URL <http://localhost/static1/statique-1&dep=26>. Ici la page affiche bien le numéro du département. D'où provient le problème ?

Le fichier qui s'occupe de convertir les URLs réécrites en URLs classiques de PluXml (`index.php?static1/statique-1`) est le fichier `.htaccess`.

```
RewriteRule ^(?!feed)(.*)$ index.php?$1 [L]  
RewriteRule ^feed/(.*)$ feed.php?$1 [L]
```

La première règle indique que les URLs (hors nom de domaine) ne commençant pas par `feed` doivent être intégralement recopiées à la suite de `index.php`. C'est ce qui se passe dans la plupart des cas.

Par exemple, l'URL <http://localhost/static1/statique-1> est convertie en <http://localhost/index.php?static1/statique-1> sans que cela ne pose de problème. Ici le nom de la page est donc traité comme un paramètre. Que ce passe-t-il avec nos deux URLs :

<http://localhost/static1/statique-1&dep=26>

Ici, la fin de l'URL est intégralement recopiée à la suite du `/index.php?`, ce qui donne <http://localhost/index.php?static1/statique-1&dep=26>, cela fonctionne bien car nous avons bien, le nom de domaine, le chemin, le nom du fichier, le ? qui indique le début des paramètres, et deux paramètres séparés par un &.

Voyons la seconde URL :

<http://localhost/static1/statique-1?dep=26>

Cette URL est convertie en <http://localhost/index.php?static1/statique-1> le dep=26 est donc perdu.

Le paramètre a été ignoré, car, par défaut, les règles de réécriture de .htaccess suppriment tous les paramètres. Il faut indiquer à la règle de recoller les paramètres à la suite de l'URL lorsque la conversion est faite.

Nous utilisons donc le flag QSA qui sert à replacer le dep=26 à la fin de l'URL, une fois la conversion effectuée.

Nos lignes deviennent donc

```
RewriteRule ^(?!(feed) (.*)$ index.php?$1 [QSA,L]
RewriteRule ^feed\/(.*)$ feed.php?$1 [QSA,L]
```

Les deux URLs fonctionnent dorénavant. Il est conseillé d'utiliser <http://localhost/static1/statique-1?dep=26> qui est la méthode standard pour passer des paramètres.

Nous avons, par la même occasion corrigé (partiellement) le problème de la non-validité du &, puisqu'il n'est plus utilisé. Mais dans le cas où on a plusieurs paramètres, l'utilisation de & est inévitable.

II. La non validité de & dans le code d'un lien

Considérons cet extrait de code :

```
<a href="http://localhost/static1/statique-1&dep=06">Lien 1</a>
```

Il n'est pas valide car il contient le caractère & qui sert normalement à indiquer le début d'un caractère spécial (, <...). Ici ce caractère est utilisé en temps que tel. Il faut donc utiliser sa propre séquence d'échappement : &.

Cela paraît simple avec un code statique, mais la fonction `$plxShow->plxMotor->urlRewrite` réagit mal aux passages d'un & ; en paramètre, ce qui nous complique un peu la tâche.

Exemple :

```
echo $plxShow->plxMotor->urlRewrite('http://localhost/index.php?
static1/statique-1&dep=06');
```

 fonctionne très bien, mais renvoie une URL contenant un &, donc non valide.

Au contraire, `echo $plxShow->plxMotor->urlRewrite('http://localhost/index.php?
static1/statique-1&dep=06');` renvoie une URL non fonctionnelle (<http://localhost/static1/statique-1&#amp;dep=06>).

Il faudrait modifier la fonction `$plxShow->plxMotor->urlRewrite` afin qu'elle échappe elle-même les `&` pour éviter de lui faire convertir des liens contenant des `&`.

Une solution

```
/**
 * Méthode qui réécrit les urls pour supprimer le ?
 *
 * @param url url à réécrire
 * @return string url réécrite
 * @author Stéphane F
 */
public function urlRewrite($url='') {

    if($url=='') return $this->racine;

    preg_match('/^([0-9a-z\_\-\.\/]+)?[\?]?([0-9a-z\_\-\.\/,&=%]+)?[\#]?(.*)
$/i', $url, $args);

    if($this->aConf['urlrewriting']) {
        $new_url = str_replace('index.php', '', $args[1]);
        $new_url = str_replace('feed.php', 'feed/', $new_url);
        $new_url .= !empty($args[2])?$args[2]:'';
        if(empty($new_url)) $new_url = $this->path_url;
        $new_url .= !empty($args[3])?'#'.$args[3]:'';
        $url = $this->racine.$new_url;
    } else {
        if(empty($args[1]) AND !empty($args[2])) $args[1] = 'index.php';
        $new_url = !empty($args[1])?$args[1]:$this->path_url;
        $new_url .= !empty($args[2])?'?'.$args[2]:'';
        $new_url .= !empty($args[3])?'#'.$args[3]:'';
        $url = $this->racine.$new_url;
    }
    return str_replace('&', '&amp;', $url);
}
```

En conclusion, les premières pistes pour améliorer les URLs de PluXml sont là, mais elles n'ont bien sûr pas été testées en détail mais mériteraient de l'être.