Xaraya
Request for Comments: 0023
Category: Informational

M.
Xaraya Development Group

July 2002

Short URL Support

Status of this Memo

This memo provides information for the Xaraya community. It does not specify an Xaraya standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

The contents of this RFC contain the literal content of the old plain text version of RFC-0023

When time is a less scarcer good, someone might convert the plain text into structured XML so we can benefit from it.

Table of Contents

1. Introduction

The following RFC is a first summary and describes some solution proposals based on the articles on search-engine-friendly URLs at http://www.postnuke.com/, on discussions regarding user-friendly URLs in the xardev mailing list, and on a test implementation by mcanini and mikespub for the articles module in the .8 branch. All ?xxx numbers refer to the basic documents in section 7.

2. List of requirements for Short URL Support

Short URL support should cover 2 important (but slightly different) goals :

* user-friendly URLs : for exchanging URLs, manual linking and more intuitive navigation inside a Xaraya site

* search engine-friendly URLs : for facilitating crawling/spidering by robots, and automated linking by search engines, newsfeeds etc.

In both cases, it should be possible to convert the current URL format http://www.mysite.com/index.php?module=...(&type=...)&func=...&pid=... into a virtual URL in the style http://www.mysite.com/index.php/.../.../...(?...)

For user-friendly URLs, the resulting virtual URLs should also be "semantically meaningful" where possible. [2]

In addition, the following technical requirements apply :

* not only for Apache webservers

This means that short URL support shouldn't require mod_rewrite

* minimal impact on performance

This means that short URL support shouldn't rely on some global search & replace of any URLs found in the HTML output buffer right before it's sent to the browser

* site administrators should be able to turn on/off short URLs, This means providing configuration settings for the site admins

3. Solution proposals - database tables

None needed at the moment.

A global configuration variable 'System.UseShortURLsIfPossible' specifies if short URLs should be used where possible or not.

A module-specific variable 'SupportShortURLs' specifies if a particular module can support short URLs or not.

A possible future version might support dynamic mapping of paths to parameters (and vice-versa), at which point a mapping table might become useful. See section 10 for details.

4. Solution proposals - functions

4.1 Design used for the test implementation

The Xaraya API provides two unique places where URLs are respectively generated and interpreted : xarModURL() and xarGetRequestInfo(). This means that any modification of the URL format can be limited to those two functions, for any module supporting the Xaraya API.

This gives us a very easy way to start using short URLs, without affecting anything else from the Xaraya core or modules.

The one major open issue for the test implementation was how to make sure we can generate short URLs that fit the 2 design goals mentioned above.

If we only aim for search engine-friendly URLs, we might use some automatic conversion routine in those two functions, and be done with it. However, if we aim for user-friendly URLs as well, *someone* has to determine what might be meaningful URLs for a particular module - and that someone is generally the module developer (or someone customising that module to his own taste).

For the test implementation, we have therefore chosen the following URL scheme for short URLs :

http://www.mysite.com/index.php/<modulename>/<something module-specific>

In practice, a further distinction is made between user-related functions and URLs, and admin-related functions and URLs :

index.php/<modulename>/<something defined in xaruserapi.php>, and

index.php/<modulename>/admin/<something defined in xaradminapi.php>

(Note : in the short term, we don't expect to see many short URLs for admin-related links, but allowing this distinction from the start makes things easier to plan for)

In order to generate a short URL for a particular set of module parameters, xarModURL() will call the function <module>_<type>_encode_shorturl, pass it the module function and parameters to be encoded, and expect back a virtual path for the short URL. If no path is returned, this means this particular combination of parameters doesn't have an equivalent short URL, e.g. because there is no reasonable meaningful short URL for it.

Example : selecting articles that are in a particular category could easily be given a short URL in the style /articles/category, but selecting articles that belong to several categories at the same time may not really have a meaningful equivalent short URL.

In the other direction, xarGetRequestInfo() will call the module-specific <module>_<type>_decode_shorturl() function, pass it the virtual path and expect back an array with the corresponding module parameters. Again, there may not be a reasonable equivalent, e.g. if a user tries to play a bit with the short URL.

4.2 <module>_<type>_encode_shorturl() function

The following example is taken from modules/articles/xaruserapi.php, and generates the following short URLs :

index.php/articles/index.html (default index view)

index.php/articles/NN.html (view a particular article)

index.php/articles/<pubtype>/index.html (e.g. news, sections, ...)

index.php/articles/categoryNN.html (e.g. topics)

index.php/articles/map.html (overview map of publication types and categories)

Pagination is supported via an optional ?startnum=NN parameter.

Obviously, this URL scheme can be improved later on...

/** * return the path for a short URL to xarModURL for this module * @param $args the function and arguments passed to xarModURL * @returns string * @return path to be added to index.php for a short URL, or empty if failed */

```
function articles_userapi_encode_shorturl($args) {

    // Get arguments from argument array
    extract($args);

    // check if we have something to work with
    if (!isset($func)) {
        return;
    }

    // make sure you don't pass the following variables as arguments too

    // default path is empty -> no short URL
    $path = '';
    // if we want to add some common arguments as URL parameters below
    $join = '?';
    // we can't rely on xarModGetName() here (yet) !
    $module = 'articles';

    $pubtypes = xarModAPIFunc('articles','user','getpubtypes');

    // specify some short URLs relevant to your module
    if ($func == 'main') {
        $path = '/' . $module . '/index.html';
    } elseif ($func == 'view') {
        if (isset($cids) && is_array($cids)) {
            if (count($cids) == 1) {
                $path = '/' . $module . '/category' . $cids[0] . '.html';
            } else {
                // this is *not* supported for short URLs -> do nothing
                // (e.g. for multiple category selections)
            }
        } elseif (isset($ptid) && isset($pubtypes[$ptid])) {
            $path = '/' . $module . '/' . $pubtypes[$ptid]['name'] .
'/index.html';
        } else {
            $path = '/' . $module . '/index.html';
        }
    } elseif ($func == 'display' && isset($aid)) {
        $path = '/' . $module . "/$aid.html";
    // TODO: do we even care about those when we display an article ?
        // just add the cids as regular parameters ?cids[0]=1&cids[1]=4...
        //if (isset($cids) && is_array($cids)) {
        //    foreach ($cids as $id => $cid) {
        //        if (isset($cid)) {
        //            $path .= $join . "cids[$id]=$cid";
        //            // change the join character (once would be enough)
        //            $join = '&';
        //        }
        //    }
        //}
    } elseif ($func == 'viewmap') {
        $path = '/' . $module . '/map.html';
    }
    // anything else does not have a short URL equivalent

    // add some other module arguments as standard URL parameters
```

```
    if (!empty($path) && isset($startnum)) {
        $path .= $join . 'startnum=' . $startnum;
    }

    return $path;

}
```

4.3 <module>_<type>_decode_shorturl() function

The following example is taken from modules/articles/xaruserapi.php, and in addition to the short URLs mentioned above, it also supports URLs like :

index.php/articles

index.php/articles/<pubtype>

in case a user feels like "browsing" your site via the URLs.

/** * extract function and arguments from short URLs for this module, and pass * them back to xarGetRequestInfo() * @param $params array containing the elements of PATH_INFO * @returns array * @return array containing func the function to be called and args the query * string arguments, or empty if it failed */

```
function articles_userapi_decode_shorturl($params) {

    $args = array();
    if (empty($params[1])) {
        return array('main', $args);
    } elseif ($params[1] == 'index.html') {
        return array('main', $args);
    } elseif ($params[1] == 'map.html') {
        return array('viewmap', $args);
    } elseif (preg_match('/^(\d+)\.html$/',$params[1],$matches)) {
        $aid = $matches[1];
        $args['aid'] = $aid;
        return array('display', $args);
    } elseif (preg_match('/^category(\d+)\.html$/',$params[1],$matches)) {
        $cids[0] = $matches[1];
        $args['cids'] = $cids;
        return array('view', $args);
    } else {
        $pubtypes = xarModAPIFunc('articles','user','getpubtypes');
        foreach ($pubtypes as $id => $pubtype) {
            if ($params[1] == $pubtype['name']) {
                $args['ptid'] = $id;
                return array('view', $args);
            }
        }
    }

    // default : return nothing -> no short URL
    // (e.g. for multiple category selections)

}
```

4.4 Relationship to other areas

Since short URL support can be enabled on a module-by-module basis, older modules can still continue working as before while newer modules can start supporting short URLs. No other areas should be affected.

One possible extension could be a better use of caching by intermediate cache devices, proxies etc. For this, further optimisation of Xaraya performance by using Last-Modified headers would definitely be a good idea.

4.5 Code that will need to be rewritten

In the test implementation, the following functions were adapted :

4.5.1 xarModURL() in includes/xarMod.php (encoding)

This function now checks if 'System.UseShortURLsIfPossible' is set, then checks if 'SupportShortURLs' is set for that particular module, and then tries to call xarModAPIFunc($modname, $type, 'encode_shorturl', $args) in order to encode the parameters into a short URL.

4.5.2 xarGetRequestInfo() in includes/xarAPI.php (decoding)

If no module is specified in the 'regular' way and there is a virtual path appended to index.php, this function now performs the same checks as xarModURL() and then calls xarModAPIFunc($modname, $type, 'decode_shorturl', $params) in order to decode the virtual URL path into parameters. The resulting parameters are stored in $xarAPI_?RequestVariables for further processing.

4.5.3 xarVarCleanFromInput() in includes/xarAPI.php (decoding)

This function now also checks if the requested variable is found in $xarAPI_?RequestVariables.

4.5.4 xarModGetAlias(), xarModSetAlias, xarModDelAlias() in includes/xarMod.php (both)

These functions were added to allow module developers / site admins to define aliases for module names, in the context of short URL support.

For example, /index.php/news/ can be used instead of /index.php/articles/news/ by declaring 'news' to be an alias for the 'articles' module in the articles admin interface. In *_encode_shorturl, we can check if 'news' is a known alias, and if so, generate the /index.php/news/ URL instead of the longer alternative. And in *_decode_shorturl, we can check $params[0] to see if some special alias was used, and if so, preset the right publication type corresponding to it.

In addition, the administration of the base module was adapted to allow site administrators to set/unset the 'System.UseShortURLsIfPossible' variable.

4.6 Basic documents for this RFC

[1]
http://centre.ics.uci.edu/~grape/modules.php?op=modload&name=Wiki&file=index&pagename=Search%20Engine%20Optimiza

Credits to besfred

[2] URL as UI : http://www.useit.com/alertbox/990321.html

Credits to Jakob Nielsen

[3] Fighting Linkrot : http://www.useit.com/alertbox/980614.html Credits to Jakob Nielsen

TODO: add links to search engine articles at http://www.postnuke.com/ and user-friendly URLs in xardev mailing list

4.7 Tools that need to be created from scratch

None yet. A "convertor module" for transforming incoming requests for old-style URLs (pre-.71x) to new-style URLs (short or not) could be very useful for sites that are already heavily linked to, and that are migrating to .8+ [3] This issue is not specific for short URL support, though.

4.8 Current Issues

Explore willingness of module developers to have their own encoding/decoding routines in order to provide meaningful URLs.

Define some examples and recommendations for meaningful URL schemes. Example : .../year/2002.html, .../month/2002-07.html, .../mycategory, ...

5. Retractions

The following features were considered but not retained for the .80 test implementation. They might be re-considered for future .8x or .9x versions.

5.1 Support multi-lingual short URLs

E.g. allow different virtual URLs depending on the language.

This can already be done for everything *after* the module name in the URL, if the _encode_shorturl and _decode_shorturl handle the ML aspect, but not for the module name itself. With the use of module aliases, the module name part could also be changed in the URL, but there is no administration interface for this 'trick'.

5.2 Support non module-related first parts in the URL

E.g. use a virtual path for showing everything related to a certain category, or for some other page showing output from multiple "main modules".

This should probably wait until BL supports multiple main modules, or some other way is found to generate non-module-related output (independently of whether we're using short URLs or not).

5.3 Allow site administrators to customize the short URLs

E.g. have some administration interface when site admins can change the mapping between virtual URLs and the corresponding module parameters.

This isn't really feasible with the current test implementation, but dynamic mapping (in the _encode_shorturl and _decode_shorturl functions, and/or elsewhere) could be implemented at a later stage.

In the meantime, the paths generated or interpreted by the different _encode_shorturl and _decode_shorturl functions can of course be modified directly in the code by the site admins.

5.4 Get rid of the /index.php/ part of the virtual URL

The index.php (or at least *some* PHP script) is necessary if you don't want to (or can't easily) change your webserver configuration.

Doing a straight mapping of /articles to /index.php/articles (or to /index.php?module=articles or whatever) is not something Xaraya can handle - it must be done by the webserver *before* the request even gets handed over to Xaraya...

For people who have control over the configuration of their own website, you can get rid of that index.php by defining virtual paths in your webserver - but that would require using mod_rewrite again, or mapping something like /xarsite to /index.php, or some similar server-side mapping.

6. Changelog

1.1 (August 27, 2002) mikespub

Add module alias functions

1.0 (July 23, 2002) mikespub

Initial Version