

Comments API

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

Copyright © The Digital Development Foundation (2002). All Rights Reserved.

Abstract

This RFC contains a proposal for the implementation of a comments API.

Table of Contents

| | |
|--|-----------|
| 1 Introduction..... | 3 |
| 2 List of requirements for the Comments API..... | 4 |
| 3 Solution proposals - database tables..... | 5 |
| 3.1 Table Comment..... | 5 |
| 4 Solution proposals - functions..... | 6 |
| 4.1 Backend Code..... | 6 |
| 4.2 Rendering Code..... | 8 |
| 5 Relationship of the API to other areas..... | 10 |
| 6 Code that will need to be rewritten..... | 11 |
| 7 Tools that need to be written from scratch..... | 12 |
| 8 Current Issues..... | 13 |
| 9 Retractions..... | 14 |
| 10 Changelog..... | 15 |
| 11 References..... | 16 |
| Author's Address..... | 17 |
| Intellectual Property and Copyright Statements..... | 18 |

1. Introduction

All ?xxx numbers refer to the basic documents in section 7. We read through the articles on <http://www.postnuke.com>, its forum threads, feature requests on sourceforge and the developers mailing list. The following RFC is a first summary and contains some solution proposals either compiled from the basic documents or from general definitions concerning comments systems / forums.

2. List of requirements for the Comments API

The main objective of this API is to provide a small, yet flexible codebase for adding comment functionality to modules. This API will not replace a full-blown forum, but will rather aim to implement core functionality. A forum could conceivably make use of this API to extend on it and provide advanced functionality. It might be feasible to make parts of this API pluggable to allow for the addition of more rendering types for instance.[\[7\]](#) Following is a list of requirements broadly grouped into categories.

1. Backend
2. Keep the codebase small to make maintenance easier
3. Separate display and core functionality code
4. The API should be completely transparent for the modules, and vice versa. There should be no dependencies between a module and the API.
5. Provide gateway functions to
 - Email
 - NNTP

Authors of posts can opt to receive replies to their posts via email. Maybe allow others to subscribe to a thread. Note that this is closely related to a generic Notification API. See section 5 for more information
6. Make comments url-addressable [\[4\]](#). This allows to refer to individual comments with hyperlinks.
7. Provide several search & summary functions, like the date of the last comment for a given discussion [\[10\]](#).
8. Moderation
9. Rely on permissions system for moderation access [\[1\]](#)
10. Implement a bot posting prevention system [\[3\]](#), [\[14\]](#)
11. Allow the deletion of subcomments. [\[5\]](#) This raises a lot of questions what should happen to child comments if a parent comment is deleted.
12. Rendering
13. Provide pluggable rendering modules, maybe from other forums [\[2\]](#) The comments API will ship with a few basic rendering modules such as threaded and flat views.
14. Implement pagination so that long lists of comments do not overwhelm the layout.

3. Solution proposals - database tables

This is just one proposal for a table structure, the one chosen by the XOOPS project can be found in [21].

3.1 Table Comment

```
// as the comment repository
commentid      unique long integer value
modname        holds a reference to the module (see section 5g)
hookid         references an item in the module, semantics are
               defined by the module.
author         foreign key to the users table
title          the title of the comment
comment       the comment, maximum size to be determined
comdate        the date of the comment as UTC timestamp
hostname       records the hostname of the poster
lineage        the family tree upto and including the current
               comment
```

4. Solution proposals - functions

This is just one proposal for an implementation, the one chosen by the XOOPS project can be found in [20].

4.1 Backend Code

The Backend API gives the developer a means in which s/he can seamlessly interact with the database. All database interaction is completely handled by the following routines. List of Functions:

```
xarComments_Get(modname, hookid, CommentId=0, depth=-1, sort='ASC')
xarComments_Add(modname, hookid, ParentId, author, title, comment, hostname)
xarComments_Delete(CommentId, delkids=true)
xarComments_Modify(CommentId, title, comment)
xarComment_Search(searchstring [,module] [,hook]);
xarComment_Stats([module, [hook]]);
xarComments_MoveThread(AttachToId, CommentId, IncludeParent=true);
```

Details of Functions:

In these functions you will notice the parameter 'hook' -- this parameter is the itemid of the module in question (ie., the 'hook' into the module).

Common Function Parameters:

```
$module:  string (or integer), module name or possibly
          the registered id of the module.
$hook:    integer, itemid of the module. (ie., for
          News module it would be the storyid)
```

Common Return:

The initial proposal is to have a named array for the comments, the structure of the array is shown below:

```
commentarray(
"CommentId" => 423,
"author" => "Joey Butta",           // name of the author
"title" => "Booga Booga",           // title of the comment
"date" => "(timestamp)",           // date the comment was made
"comment" => "Nothing to say.",     // the comment itself.
"depth" => "3",                     // level depth in the parent/child
                                     // tree
"hostname" => "192.168.1.9",        // hostname or ip of the author
"lineage" => "23:64:234:322:328:423", // "family tree" for the comment
"haskids" => false
);
```

Function Descriptions:

```
FUNCTION: xarCommentGet
RETURN: commentarray OR array of commentarrays
PARAMETERS:
    $module:    see above;
    $hook:      see above;
               $sort:    string ("ASC" or "DESC")
    $commentid: integer (0 or greater)
    $depth:     integer (-1, 0 or >0)
```

Acquires comments based on the module/hook and the comment id. The depth parameter allows us to only

acquire comments to (n) levels deep. A value of 0 would acquire the sole comment specified by the comment id and -1 would acquire all comments down to the deepest level. If both the commentid and the level are omitted from the function call (ie., `xarCommentGet('News','1')`), all comments for the given module/hook are retrieved. To retrieve a single comment the call would be: `xarCommentGet('News',1,5,0)` 1 being the storyid, 5 being the commentid, and 0 telling the function to only return that one comment. If sort is specified, the top most parents are sorted by date in the direction specified, otherwise they are sorted in ascending order by date.

```
FUNCTION: xarCommentGetKids
RETURN: commentarray OR array of commentarrays
PARAMETERS:
  $module:      see above;
  $hook:        see above;
  $parent:      integer (1 or greater)
  $level:       integer (0 or greater)
```

Acquires a list of comments based on a particular thread. This is similar to the current `?DisplayKids(...)` function. The level parameter keeps track of how many levels deep in the the parent/child tree we are. This function is intended to be nested and will be utilized in the function `xarCommentGet(...)` as a means of acquiring comments down to a particular depth. If there are no children for the particular comment, an empty array is returned.

```
FUNCTION: xarCommentAdd
RETURN: bool (True or False)
PARAMETERS:
  $module: see above;
  $hook:   see above;
  $commentarray: named array
```

Adds a Comment to the database based on the module/hook and returns a true or false on success. The `commentarray` contains the actual comment.

```
FUNCTION: xarCommentDel
RETURN: bool (True or False)
PARAMETERS:
  $module: see above;
  $hook:   see above;
  $commentid: integer (0 or greater)
```

Deletes comment(s) from the database. If the comment has children and `boolchildren` is false, all first level children's parent ids are reset to the grandparent's id or zero if the parent is a toplevel comment. Note: this does not affect the relationship between the first level children and any children they might have. Otherwise, if `boolchildren` is set to true, the comment and `=all=` it's children are deleted from the database.

```
FUNCTION: xarCommentModify
RETURN: bool (True or False)
PARAMETERS:
  $module: see above;
  $hook:   see above;
  $commentid: integer (0 or greater)
  $commentarray: named array
```

Modifies a comment in the database based on the module/hook/commentid. The `commentarray` contains the actual comment.

```
FUNCTION: xarCommentSearch
RETURN: commentarray OR array of commentarrays
PARAMETERS:
  $module: see above;
  $hook:   see above;
  $searchstring: string
```

Returns a comment or array of comments based on the search string. If the module/hook is given, it searches only the specific module/hook for the searchstring. If the module is given and not the hook, it searches for all comments matching the searchstring for the given module only, likewise if the hook is given without the module. If both the module/hook are not given, it returns ALL comments. Note The comment array structure returned by this function is slightly different as it would include the module/hook as key elements.

```
FUNCTION: xarCommentNext
RETURN: commentarray
PARAMETERS:
    $commentarray: named array
```

Returns the next comment in the comments array - this function and it's counterpart, `xarCommentPrev(...)` are helper functions similar in concept to `adodb's ?MoveNext()` function. Each time this function is called, the array pointer is moved ahead one comment traversing all branches (ie., children) of the tree one at a time until the end is reached (ie., the last comment).

```
FUNCTION: xarCommentNext
RETURN: commentarray
PARAMETERS:
    $commentarray: named array
```

returns the previous comment in the comments array.

```
FUNCTION: xarCommentStats
RETURN: statsarray
PARAMETERS:
    $module: see above;
    $hook:   see above;
```

Returns an array with last comment, date of last comment, # of comments etc

4.2 Rendering Code

Rendering will be done through the use of a class factory. This allows for the implementation of different renderers without changing the API. Each renderer has the exact same methods.

List of Functions

```
xarCommentRenderFactory($type) <-- Constructor
Class Renderer
Renderer() <-- Constructor
xarCommentDraw($commentarray, $optionsarray)
xarCommentDrawInput()
    xarCommentDrawEdit($commentid)
```

Details of Functions

Common Function Parameters

```
$commentarray  This array holds the comments to be drawn. See
                above for its structure.
$optionsarray  This array holds the renderer-specific options
                in key->value form.
```

Common Return

All functions in the rendering code return XHTML in a variable.

Function Descriptions

```
FUNCTION: xarCommentRenderFactory
RETURN: renderer class
PARAMETERS:
    $type: the name of the renderer to use.
```

returns the chosen renderer class.

```
FUNCTION: xarCommentDraw
RETURN: XHTML
PARAMETERS:
    $commentarray: named array.
    $optionsarray: named array.
```

returns the comments specified rendered in XHTML.

```
FUNCTION: xarCommentDrawInput
RETURN: XHTML
PARAMETERS:
```

returns the comment input form rendered in XHTML.

```
FUNCTION: xarCommentDrawInput
RETURN: XHTML
PARAMETERS:
    $commentid: id of the comment to edit.
```

returns the comment edit form for the specified comment rendered in XHTML

5. Relationship of the API to other areas

Textsanitizer

The comments should go through a foul language filter before being posted. That filter should not be part of the comments system though. Also, support for various text encoding systems (HTML, bbcode, wikicode) will be provided by the Textsanitizer.

Permissions

The Comments API will make extensive use of the Permissions system to implement access control and the display of elements [6]. The option to turn comments on or off for a given module [9] will also be handled by the Permissions system. It remains to be seen whether extensive use of the Permissions system will lead to an overload within that system.

Categories

It might be beneficial to attach metadata to comments via the Categories system.

Users

What should happen with orphaned content? [11]

Hooks to external editors

Spell checkers [13] and WYSIWYG editors will be hooked in via the standard methods.

Notification systems

Several aspects of the Comments API would benefit from plugging into a Notification system. This would regularly update subscribed parties on the status of comments.

Module system

The Comments API needs a reliable way to refer to modules. Currently this issue remains unresolved. [15]

6. Code that will need to be rewritten

```
/modules/Downloads  
/modules/News  
/modules/NS-Comments  
/modules/Polls  
/modules/Reviews  
/modules/Search  
/modules/Sections  
/modules/Top_List  
/modules/Topics  
/modules/Web_Links  
and some blocks
```

7. Tools that need to be written from scratch

A helper function to determine the real IP address of a user. This function can make use of the Forwarded-for header. [\[16\]](#)

8. Current Issues

Current proposal for retrieving comments from the database is very expensive Possible fix: encode the parent/child hierarchy in the key. [\[17\]](#)

9. Retractions

We list features that were considered but rejected for this API below.

1. Let anonymous users specify a name [\[12\]](#) The reasoning was that we do not want to establish a secondary way of registering with Postnuke, an account light of sorts. Experience with live sites has shown that appending the name to a post works well.
2. Implement core functionality for a moderation system. There are lots of different ways to implement moderation systems: Karma (ala slashdot.org), Network of Trust (ala advogato.org) [\[8\]](#). Provide a simple field to record the score of a comment regardless of the means by which this score was calculated. ?Not implemented due to upcoming Rating System RFC/API

10. Changelog

```
4.1 (January 7, 2002) modified by Carl P. Corliss (aka Rabbitt)
  Added 'haskids' field to db table scheme
  Added function xarCommentSearch to Section 4a
  (Function List and Description)
  Modified function xarCommentGet -- added parameter 'sort'
4.0 (January 6, 2002) modified by Gregor J. Rothfuss
  Section 4b extended.
  Formatting cleanups.
3.5 (January 6, 2002) modified by Carl P. Corliss (aka Rabbitt)
  Sections 4a, 4b & 9 added.
  #2 of Section 10:Retractions moved from Section 5:Moderation.
  'rating' removed from comment table layout.
3.0 (January 4, 2002) modified by Gregor J. Rothfuss
  Section 8 added.
2.0 (January 4, 2002) modified by Gregor J. Rothfuss
  Database schema added.
  Section 5g added.
1.0 (January 3, 2002)
  Initial Version
```

11 References

- [1] Jim McDonald, " <http://groups.yahoo.com/group/pndev/message/5845>", unknown.
- [2] chv, hzink, " <http://groups.yahoo.com/group/pndev/message/5791>", unknown.
- [3] electric monk, " <http://groups.yahoo.com/group/pndev/message/5465>", unknown.
- [4] goosepro, " http://sourceforge.net/tracker/index.php?func=detail&aid=485710&group_id=27927&atid=392231", unknown.
- [5] unknown, " http://sourceforge.net/tracker/index.php?func=detail&aid=456931&group_id=27927&atid=392231", unknown.
- [6] unknown, " http://sourceforge.net/tracker/index.php?func=detail&aid=477620&group_id=27927&atid=392231", unknown.
- [7] unknown, " http://sourceforge.net/tracker/index.php?func=detail&aid=484304&group_id=27927&atid=392231", unknown.
- [8] unknown, " http://sourceforge.net/tracker/?group_id=27927&atid=392231&func=detail&aid=440629", unknown.
- [9] unknown, " http://sourceforge.net/tracker/?group_id=27927&atid=392231&func=detail&aid=442294", unknown.
- [10] unknown, " http://sourceforge.net/tracker/?group_id=27927&atid=392231&func=detail&aid=442947", unknown.
- [11] unknown, " http://sourceforge.net/tracker/?group_id=27927&atid=392231&func=detail&aid=445490", unknown.
- [12] unknown, " http://sourceforge.net/tracker/?group_id=27927&atid=392231&func=detail&aid=455164", unknown.
- [13] unknown, " http://sourceforge.net/tracker/?group_id=27927&atid=392231&func=detail&aid=460677", unknown.
- [14] unknown, " http://sourceforge.net/tracker/?group_id=27927&atid=392231&func=detail&aid=477934", unknown.
- [15] unknown, " <http://groups.yahoo.com/group/pndev/message/8363>", unknown.
- [16] unknown, " <http://www.cgsa.net/php/identifierShow.php>", unknown.
- [17] unknown, " <http://groups.yahoo.com/group/PN-Data/message/28>", unknown.

Author's Address

Carl P. Corliss

Xaraya Development Group

E-Mail: rabbitt@xaraya.com

URI: <http://www.xaraya.com>

Intellectual Property Statement

The DDF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the DDF's procedures with respect to rights in standards-track and standards-related documentation can be found in RFC-0.

The DDF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the DDF Board of Directors.

Acknowledgement

Funding for the RFC Editor function is provided by the DDF