

## CUNY Academic Commons - Feature #5487

### Improvements to Social Paper group directories

2016-04-21 10:03 PM - Boone Gorges

<b>Status:</b>	Resolved	<b>Start date:</b>	2016-10-31
<b>Priority name:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Daniel Jones	<b>% Done:</b>	100%
<b>Category name:</b>	Social Paper	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.10		
<b>Description</b>			
From <a href="https://redmine.gc.cuny.edu/issues/5007">https://redmine.gc.cuny.edu/issues/5007</a> (mockups are there too):			
a. Add blue box to the top, as in Forums. Blue box will contain paper and comment count text, as well as "Create New Paper" button, but will not have "last edited by" (too complicated, and maybe not very helpful)			
b. Add pagination text (Viewing x-y of z papers) and links. Use the same format as Forums.			
c. Add Search box. Use the same format as Forums.			
Dan, you want to have a crack at these? (a) should be straightforward. Happy to help with any pointers you need on (b) or (c) - you should be able to find inspiration in how, say, bbPress does it.			
<b>Subtasks:</b>			
Feature # 6560: Search box for Social Paper group directories			<b>Resolved</b>
<b>Related issues:</b>			
Related to CUNY Academic Commons - Feature #5007: Group > Papers tab should b...		<b>Resolved</b>	<b>2015-12-09</b>

### History

#### #1 - 2016-05-09 09:07 PM - Daniel Jones

Here's my first pass a (a): <https://github.com/cuny-academic-commons/social-paper/commit/402b921c72b5c81672680ccaca048766a6522735>. Let me know what you think! One thing that seems like it would be good would be for the "new paper" link to automatically associate the new paper with the group, but I'm not sure the best way to go about that.

#### #2 - 2016-05-10 10:21 AM - Samantha Raddatz

One thing that seems like it would be good would be for the "new paper" link to automatically associate the new paper with the group, but I'm not sure the best way to go about that.

+1 for this functionality!

#### #3 - 2016-05-11 11:47 AM - Boone Gorges

Thanks, Dan! The styling looked a little off on themes other than the Commons's (the button was too big for the notices box) so I tweaked the styling in <https://github.com/cuny-academic-commons/social-paper/commit/f0fc5ed03dd1d5dbd37a43f25c6e061c8f522f6e>.

The changes are ready to test on cdev.

One thing that seems like it would be good would be for the "new paper" link to automatically associate the new paper with the group, but I'm not sure the best way to go about that.

Yeah, I'd thought the same thing. I think you can do this in JS. Something like: append `?group\_id=123` to the URL; load that into a JS variable using one of the existing calls to `wp\_localize\_script()` in the plugin; then use JS to preset the value of the 'cacsp-group-selector' field. This should only happen during paper creation (not editing), and the group\_id should probably be ignored if the current user is not a member of the specified group.

#### #4 - 2016-05-11 04:29 PM - Daniel Jones

I'm pretty close on auto-selecting the group. Running into an issue though: the link to create a new paper gets redirected, and my URL variable gets lost in the process. I can't find where that redirect happens in the code, in order to intervene to keep the query var. Any ideas?

#### #5 - 2016-05-11 05:09 PM - Boone Gorges

The redirect happens here: <https://github.com/cuny-academic-commons/social-paper/blob/master/includes/hooks-wp-fee.php#L81>. I don't know off the top of my head what's happening in the 'fee\_new' AJAX handler - it might be necessary to override that, or maybe you can maintain the query var through JS.

## #6 - 2016-05-11 06:31 PM - Daniel Jones

Thanks for pointing me in the right direction!

Here's what I ended up with: <https://github.com/cuny-academic-commons/social-paper/commit/5432aa45e62082dce1cce8b013a40c666832e7f4>

It's a little messy - I'm hoping you can let me know which pieces are likely to break and need to be beefed up.

The basic flow is:

1. Add a query arg to the new paper url
2. In the JS ajax request in hooks-wp-fee.php, use a regular expression to see if group\_id is in the original new paper url. If it is, insert it before the "#edit=true" section of the url that gets returned, or if there's no # in the returned url, just put it at the end. It seems safe to just use & without checking for earlier query args, because we'll always have one at least for the post type and ID.
3. Pass the query arg, if the user is a member of the group with that ID, to the internationalization object for a single paper
4. On the single-paper page JS, store the select2 selector in a variable, and if we have a group\_id, then use it to set the value of the selector

It seems to work on my testing, but I haven't really done weird edge cases. Do you see any red flags?

## #7 - 2016-05-11 06:32 PM - Daniel Jones

Oh and I had one type in the Javascript (checking to make sure that group\_id is in the i18n string), fixed here:

<https://github.com/cuny-academic-commons/social-paper/commit/59aad163f4d6d489fba8617da39f8448541e151c>

## #8 - 2016-05-12 03:23 PM - Daniel Jones

And here's a crack at pagination: <https://github.com/cuny-academic-commons/social-paper/commit/7d9eaec2757d103b29bec77439050d41a88df5b8>

I also fixed a couple of errors in terms of how I was counting total papers and total comments on papers in a group, from earlier commits, and made some small changes to the CSS.

## #9 - 2016-05-16 05:40 PM - Boone Gorges

These changes are looking good, Dan! I added a couple minor comments inline on the changesets.

## #10 - 2016-05-16 07:39 PM - Daniel Jones

Oh sorry about that debugging leftover! Thanks.

Yes, I see what you mean about that routine for getting the number of comments. Is there a model somewhere in the BP code for that kind of caching? Seems like there would be quite a few places I'd need to hook into to update the cached value, right? Or would it just be one place for new comment creation and one for comment deletion?

## #11 - 2016-05-23 02:32 PM - Boone Gorges

For caching, try something like this:

```
$group_paper_comment_count = wp_cache_get( $group_id, 'cacsp_group_paper_comment_counts' );
if ( false === $group_paper_comment_count ) {
    // Run the query ....
    $group_paper_comment_count = ...

    wp_cache_set( $group_id, $group_paper_comment_count, 'cacsp_group_paper_comment_counts' );
}
```

Then, for invalidation, I think you'll be able to hook into general comment insert/update/delete. Something like:

```
function cacsp_invalidate_group_paper_comment_count_cache( $comment_id ) {
    $comment = get_comment( $comment_id );
    if ( ! $comment ) {
        return;
    }

    $paper = new CACSP_Paper( $comment->comment_post_ID );

    // Not a paper.
    if ( ! $paper->exists() ) {
        return;
    }

    $group_ids = $paper->get_group_ids();
    foreach ( $group_ids as $group_id ) {
        wp_cache_delete( $group_id, 'cacsp_group_paper_comment_counts' );
    }
}
add_action( 'wp_insert_comment', 'cacsp_invalidate_group_paper_comment_count_cache' );
add_action( 'delete_comment', 'cacsp_invalidate_group_paper_comment_count_cache' );
```

```
add_action( 'edit_comment', 'cacsp_invalidate_group_paper_comment_count_cache' );
```

Untested, but this should give you an idea of the pattern.

#### #12 - 2016-06-12 04:21 PM - Daniel Jones

I haven't been able to get the cache to take, does it have something to do with this? (from the WP\_Object\_Cache class reference)

"By default, the object cache is non-persistent. This means that data stored in the cache resides in memory only and only for the duration of the request. Cached data will not be stored persistently across page loads unless you install a persistent caching plugin.

Use the Transients API instead of these functions if you need to guarantee that your data will be cached. If persistent caching is configured, then the transients functions will use the wp\_cache functions described in this document. However if persistent caching has not been enabled, then the data will instead be cached to the options table."

And for Transients:

"The Transients\_API provides persistent but temporary data caching by giving it a custom name and a timeframe after which it will be expired and regenerated.

Note: Transients only get deleted when a request is made. So, until someone visits your page and calls up the Transient, it will stay in the DB. In short: It's not a real persistent cache and not equal to stuff running on cron jobs."

And from the WP\_Object\_Cache->set() documentation:

"The \$expire parameter is not used, because the cache will automatically expire for each time a page is accessed and PHP finishes. The method is more for cache plugins which use files."

Any thoughts on what a solution could be?

#### #13 - 2016-06-12 11:01 PM - Boone Gorges

Hi Daniel - Can you say more about what "I haven't been able to get the cache to take" means? How are you testing? As you note in the documentation you reference, it's necessary to have a persistent cache backend installed in order for the cache to be persistent between page loads. If you want to be able to test this locally, install something like Memcached (google the instructions for your OS - you'll need to get the PHP extension via PECL as well as the Memcached software from your chosen repository) and then get a compatible cache drop-in for WordPress (search Github for one compatible with your PECL extension - memcache vs memcached). You could also try Redis or APCu, if they're easier for you to set up.

If you don't want to go to the trouble, you can test cache invalidation etc without a persistent cache by doing the relevant operations more than once on a given pageload. This works best in the context of unit tests, but you can also fudge it without. So:

```
global $wpdb;
$paper_comment_count = cacsp_get_group_comment_count( $group_id ); // will hit database

$num_queries = $wpdb->num_queries;

$paper_comment_count = cacsp_get_group_comment_count( $group_id ); // will not hit database
var_dump( $num_queries === $wpdb->num_queries ); // verify that this is the case

wp_insert_comment( ... );

$num_queries = $wpdb->num_queries;
$new_paper_comment_count = cacsp_get_group_comment_count( $group_id ); // will hit database, since cache has been invalidated
var_dump( $paper_comment_count + 1 === $new_paper_comment_count );
```

There are already a couple unit tests in social-paper that do something similar, which you could probably adapt fairly easily for testing. See eg test\_get\_papers\_of\_group\_cache() in tests/phpunit/tests/buddypress/groups.php. Note that since the tests run in a single pageload, they don't need a persistent cache backend.

#### #14 - 2016-06-14 10:04 PM - Daniel Jones

Thanks Boone - that was very helpful. I've got Memcached all set up and working now. Soon I'll need to get serious about unit testing.

#### #15 - 2016-06-15 02:51 PM - Daniel Jones

This is working for me locally now: <https://github.com/cuny-academic-commons/social-paper/commit/1631e8bfa0520002d5018b631a76f9c5f7a6b348>

Let me know if it looks okay, and thanks for all your help!

#### #16 - 2016-06-20 10:41 AM - Boone Gorges

Hi Dan - The cache setup looks good, but something still doesn't seem quite right. It looks to me like the \$group\_query\_not\_paged query is going to happen whether or not the cache is primed, and it's this query (rather than just looping through the results) that we're trying to avoid. Can you refactor this a bit so that the query only takes place when the cache is empty?

#### #17 - 2016-10-13 04:11 PM - Boone Gorges

Hi Dan - Could you look over this thread and see where things stand? It seems like we're pretty close.

**#18 - 2016-10-24 10:19 PM - Daniel Jones**

Thanks for flagging this Boone! I've made that fix here:

<https://github.com/cuny-academic-commons/social-paper/commit/be7b8aff52128db92ee6b186c9f771dd0ef05fc9>

To finish with this ticket, I still need to add in the search functions, I'll work on that this week.

**#19 - 2016-10-31 01:39 PM - Boone Gorges**

*- Status changed from New to Resolved*

Thanks, Dan! Too late to fit new stuff into 1.10, so I've opened [#6560](#) to track the search features.