

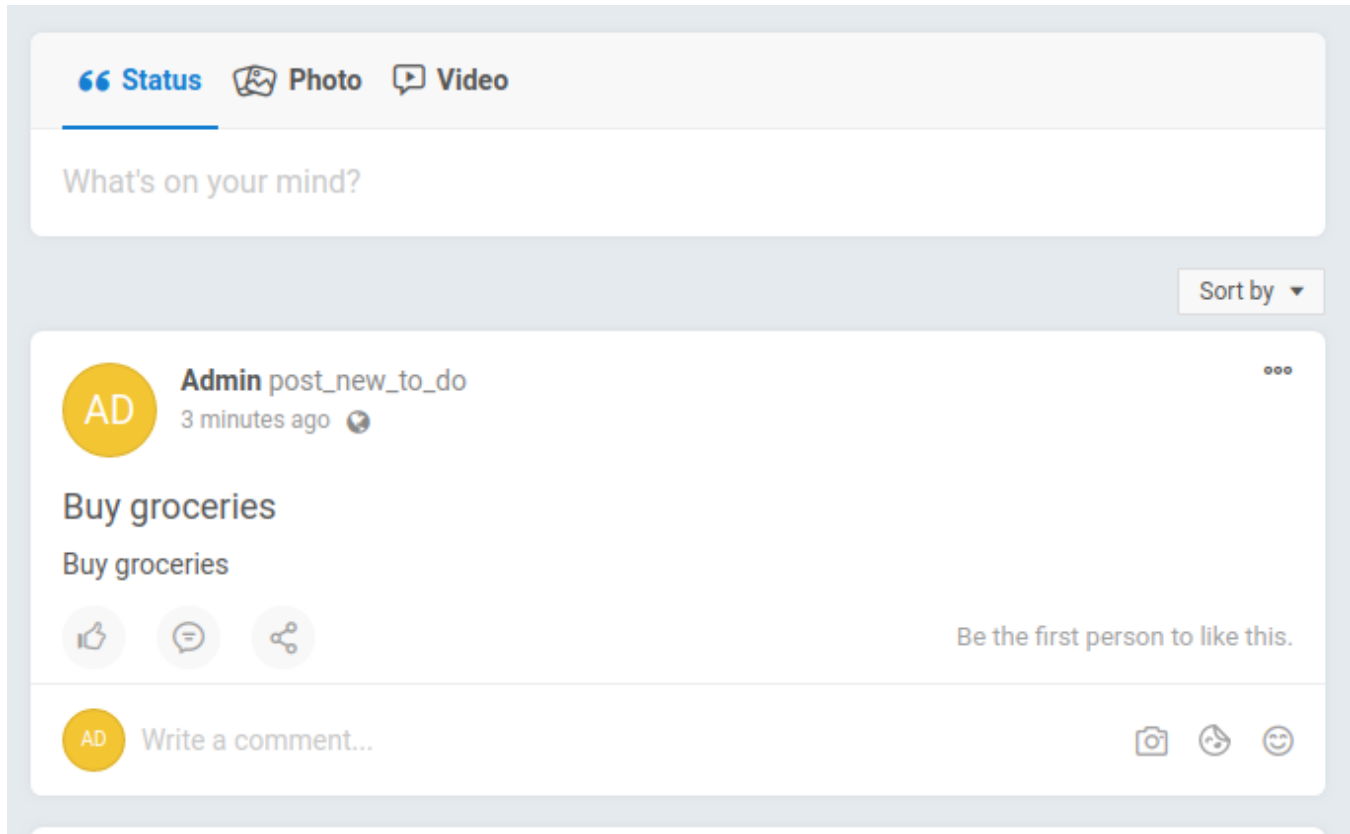
Feed Integration

Integrating Feed, Like, Comment, Share Service

What is Feed ?

In phpFox, a feed item is a story about an **actor** performs an **action** in which members can like, comment, share, ...

Example feed about admin post a status



An activity stream is a list of activities performed by a relevant actors, on phpFox system there are 2 kind of activity stream

- Stream Feeds: Showing activities performed by current logged in user, his/her friends and related pages, groups, ...
- Profile Feeds: Showing activities performed by or target to a profile, when member visit a particular profile pages (user, pages,..).

Integrating Feed Service

phpFox store feed data in table **phpfox_feed**, look in to table information, there are following important columns:

- **feed_id**: feed identity, auto increment
- **privacy**: the visibility of feed, the value of privacy is convention value, we will discuss about this later.
- **type_id**: type of feed
- **user_id**: who performed feed
- **parent_user_id**: target of feed
- **item_id**: if feed is target to pages, groups, this value contains id of object
- **time_stamp**: creation time
- **feed_reference**: where is feed is shared
- **parent_feed_id**: where is parent feed
- **parent_module_id**: f feed is target to pages, groups, this value contains type of object
- **time_update**: last modification time
- **content**: feed content
- **total_view**: total view of feed.

When a member post a new to do item, we must add a new feed with associate data

- **privacy**: 0
- **type_id**: 'todo'
- **user_id**: current logged in user id

- **item_id**: 0
- **time_stamp**: current time
- **time_update**: current time
- **feed_reference**: 0
- **parent_feed_id**: 0
- **parent_module_id**: 0

We do not insert the value directly in to database, almost case, developer using **Feed** service to handle insert process.

```
Phpfox::getService('feed.process')->add('todo', $iTaskId, 0,0);
```

Let's update **AddController.php**, modify code as following

```

<?php

namespace Apps\ToDoList\Controller;

class AddController extends \Phpfox_Component
{
    public function process()
    {
        // Get phpFox core template service
        $template = $this->template();

        // set view title
        $template->setTitle('Add To Do');

        // set view breadcrumb
        $template->setBreadCrumb('Add To Do',
            $this->url()->makeUrl('to-do-list/add'));

        // add your section menus
        $template->buildSectionMenu('to-do-list', [
            'Browse' => $this->url()->makeUrl('/to-do-list'),
            'Create' => $this->url()->makeUrl('/to-do-list/add'),
        ]);

        // get current requests
        $request = $this->request();

        // get request data
        $vals = $request->get('val');

        if (!empty($vals)) {
            // validate
            if (empty($vals['name'])) {
                \Phpfox_Error::set(_p('To do name is required'));
            }

            if (empty($vals['description'])) {
                \Phpfox_Error::set(_p('Do do description is required'));
            }

            if (\Phpfox_Error::isPassed()) {
                // insert to do item database
                $iItemId = \Phpfox::getLib('database')->insert(\Phpfox::getT('todolist_task'),[
                    'user_id'=> \Phpfox::getUserid(), // get current user id
                    'name'=> $vals['name'],
                    'description'=>$vals['description'],
                    'time_stamp'=>time(), // creation time
                    'time_update'=>time(), // last modification time
                    'privacy'=>0, // public
                    'task_status'=>0, // mark task is in-complete
                ]);

                // insert new feed
                \Phpfox::getService('feed.process')->add('todo', $iItemId, 0,0);

                $this->url()->send('to-do-list');
            }
        }
    }
}

```

Visit '+ Add To Do', add new item, visit `phpfox_feed`, table you see a new entry with `type_id` is 'todo'.

phpFox **Feed** service will handle how to fetch feed items from database, but it's does not know detail information about feed, so it will interact to others using a Callback service, in phpFox almost app have at least one callback service.

Define a callback service is simple, just add new class **Callback** under **Service** directory, past following code

```

<?php

namespace Apps\ToDoList\Service;

class Callback extends \Phpfox_Service
{
}

```

Then register callback service to **start.php**

```

<?php
// register service
$module->addServiceNames([
    'todo.browse' => Service\Browse::class,
    'todo'        => Service\ToDoList::class,
    'todo.callback' => Service\Callback::class,
]);

```

Now we will add getActivityFeed() to the Callback,

```

<?php

namespace Apps\ToDoList\Service;

use Phpfox;

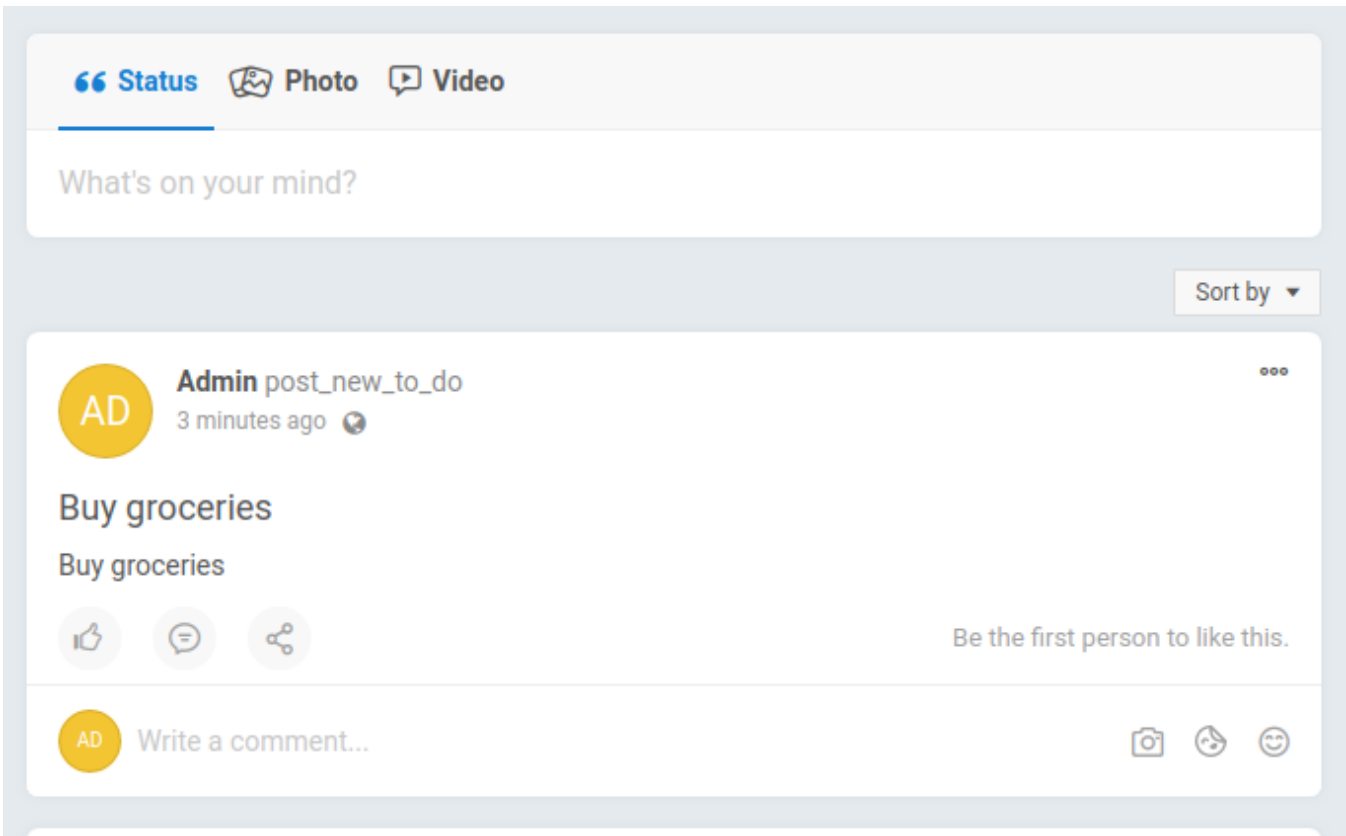
class Callback extends \Phpfox_Service
{
    public function getActivityFeed($aRow, $aCallback = null, $isChild = false)
    {
        $aItem = \Phpfox::getService('todo')->getForBrowse($aRow['item_id']);

        $aReturn = array_merge(array(
            'feed_title' => $aItem['name'],
            'privacy' => $aItem['privacy'],
            'feed_info' => _p('post_new_to_do'),
            'feed_link' => Phpfox::permalink('to-do-list.view', $aItem['task_id'], $aItem['name']),
            'feed_content' => $aItem['name'],
            'total_comment' => $aItem['total_comment'],
            'feed_total_like' => $aItem['total_like'],
            'feed_is_liked' => isset($aRow['is_liked']) ? $aRow['is_liked'] : false,
            'feed_icon' => '',
            'time_stamp' => $aRow['time_stamp'],
            'enable_like' => true,
            'comment_type_id' => 'todo',
            'like_type_id' => 'todo',
            'custom_data_cache' => $aRow
        ), $aRow);

        return $aReturn;
    }
}

```

Click on phpFox home page, you see the first feed result



Later, you can be able to change the layout of your feed.

Integrating Like Service

Continue with **Callback** Service, update code

```

<?php

namespace Apps\ToDoList\Service;

use Phpfox;

class Callback extends \Phpfox_Service
{
    // this method return required information to show a feed item on stream
    public function getActivityFeed($aRow, $aCallback = null, $isChild = false)
    {

        $aItem = \Phpfox::getService('todo')
            ->getForBrowse($aRow['item_id']);

        $isLiked = false;

        if(Phpfox::isUser()){
            $isLiked = $this->database()
                ->select('l.like_id AS is_liked')
                ->from(Phpfox::getT('todolist_task'), 'b')
                ->leftJoin(Phpfox::getT('like'), 'l', 'l.type_id = \'todo\' AND l.item_id = b.task_id AND l.
user_id = ' . Phpfox::getUserid())
                ->execute('getSlaveField');
        }

        $aReturn = array_merge([
            'feed_title'      => $aItem['name'],
            'privacy'        => $aItem['privacy'],
            'feed_info'      => _p('post_new_to_do'),
            'feed_link'      => Phpfox::permalink('to-do-list.view',
                $aItem['task_id'], $aItem['name']),
            'feed_content'   => $aItem['name'],
            'total_comment'  => $aItem['total_comment'],
            'feed_total_like' => $aItem['total_like'],
            'feed_is_liked'  => $isLiked,
            'feed_icon'      => '',
            'time_stamp'     => $aRow['time_stamp'],
            'enable_like'    => true,
            'comment_type_id' => 'todo',
            'like_type_id'   => 'todo',
            'custom_data_cache' => $aRow,
        ], $aRow);

        return $aReturn;
    }

    public function addLike($iItemId)
    {
        $this->database()->updateCount('like',
            'type_id = \'todo\' AND item_id = ' . (int)$iItemId . ',
            'total_like', 'todolist_task', 'task_id = ' . (int)$iItemId);
    }
}

```

Visit Homepage again, you can like the posted to do as the attachment:

 Status  Photo  Video

What's on your mind?

Sort by ▾



Admin post_new_to_do



8 minutes ago 

Buy groceries

Buy groceries



Like



You



Write a comment...



Add Comment Service

Continue edit **Callback** service, add 3 methods

```

<?php
// update total comment count
public function addComment($aVals)
{
    $this->database()
        ->updateCounter('todolist_task', 'total_comment', 'task_id',
            $aVals['item_id']);
}

// Comment callback required
public function getAjaxCommentVar()
{
}

// Comment callback required
public function getCommentItem($iId)
{
    $aRow = Phpfox::getService('todo')
        ->getForBrowse($iId);

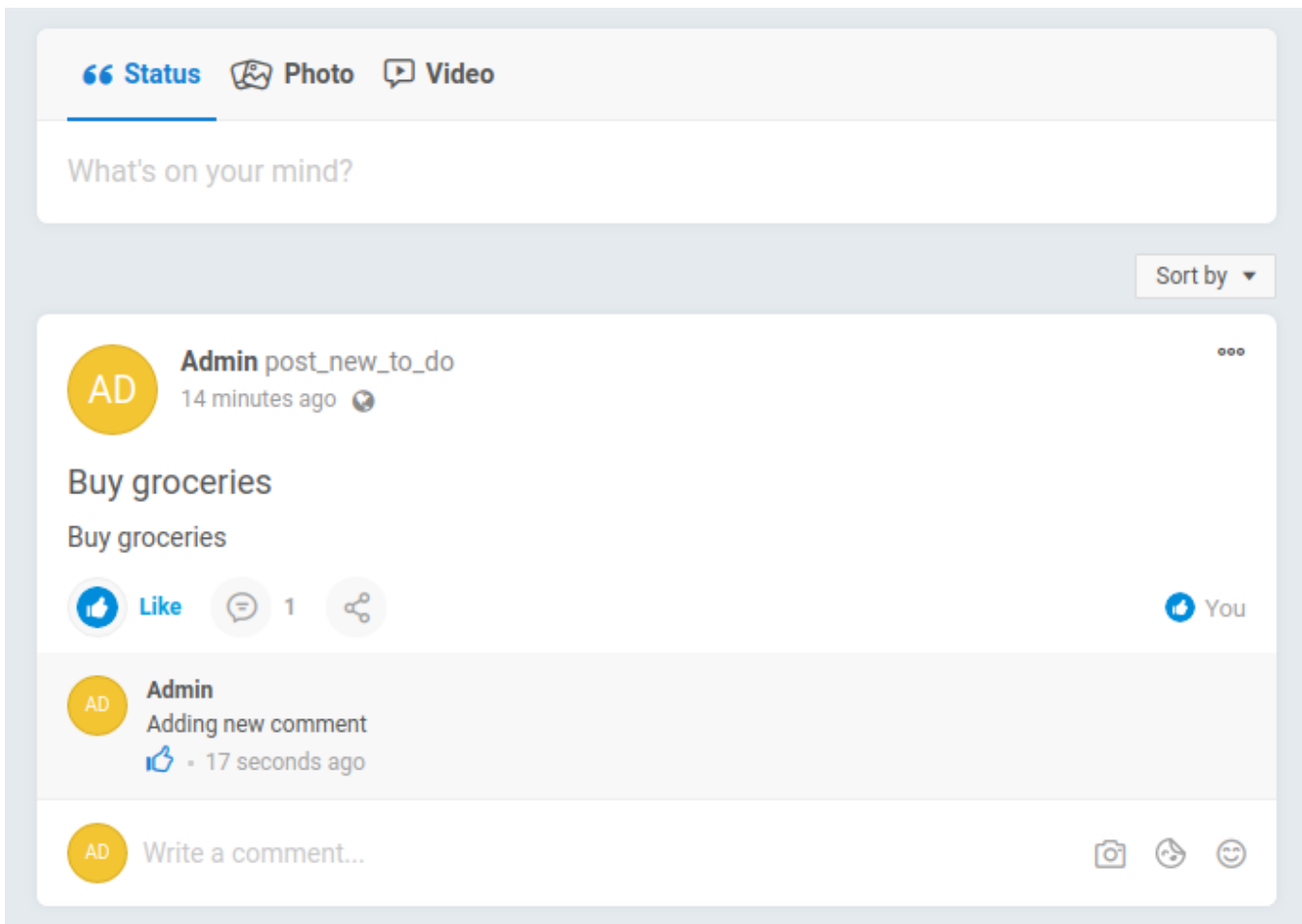
    $aRow['comment_item_id'] = $iId;
    $aRow['comment_user_id'] = Phpfox::getUserId();
    $aRow['comment_view_id'] = 0;
    if (empty($aRow)) {
        return false;
    }

    return $aRow;
}

// update total comment count
public function deleteComment($iId)
{
    $data = ['total_comment' => ['= total_comment -', 1]];
    $table = Phpfox::getT('todolist_task');
    $this->database()->update($table, $data, 'task_id = ' . (int)$iId);
}

```

Visit Home page again, you can post some comments on to do item as the attachment:



Integrating Share Service

Continue edit Callback service, add following empty method

```
// share service required callback method
public function canShareItemOnFeed(){
}
```

Visit Home page again, click on share button on to do list item, put some line then submit, the result

 Status  Photo  Video

What's on your mind?

Sort by ▾




Admin shared...

18 minutes ago 



Shared a to-do task

Admin post_new_to_do

2 hours ago 

Buy groceries

Buy groceries



Be the first person to like this.



Write a comment...



if you want to modify you feed layout, visit below chapter:

[Next Chapter](#)