

Listing To Do

The best way to fetching items with varieties of criteria is using **service**:

Put simply, a service is any PHP object that performs some sorts of "global" task. It's a purposefully-generic name used in computer science to describe an object that's created for a specific purpose (e.g. delivering emails). Each service is used throughout your application whenever you need the specific functionality it provides. You don't have to do anything special to make a service: simply write a PHP class with some code that accomplishes a specific task. Congratulations, you've just created a service!

IndexController will listing **to do items** using **Search**, **Browse** and **Pager** Service.

- **Search**: Generate search bar
- **Browse**: Control how to query database
- **Pager**: Control paginator

Register Browse Service

Create a class **Browse** under **Service** directory. then paste following code.

```
<?php

namespace Apps\ToDoList\Service;

class Browse extends \Phpfox_Service
{
    public function __construct()
    {
        $this->_sTable = \Phpfox::getT('todolist_task');
    }

    /**
     *
     */
    public function query()
    {

    }

    public function getQueryJoins($bIsCount = false, $bNoQueryFriend = false)
    {

    }

}
```

Edit **start.php**, register a new service

```
// register service
$module->addServiceNames([
    'todo.browse'=> Service\Browse::class,
]);
```

Add Search Form and Paginator

Update **IndexController**, paste following code

```
<?php

namespace Apps\ToDoList\Controller;

use Phpfox;
use Phpfox_Pager;

//controllers must be a child of \Phpfox_Component class.

class IndexController extends \Phpfox_Component
{
    public function process()
```

```

{
    // Get phpFox core template service
    $template = $this->template();

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

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

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

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

    // set is in profile
    $bIsProfile = false;

    // Configure search form
    $search = $this->search();

    // get current view
    $sView = $this->request()->get('view');

    // form action
    $sFormAction = $this->url()->makeUrl('to-do-list', ['view' => $sView]);

    $search->set([
        'type'          => 'todo',
        'field'          => 'todo.task_id',
        'ignore_blocked' => true,
        'search_tool'    => [
            'table_alias' => 'todo',
            'search'      => [
                'action'      => $sFormAction,
                'default_value' => '',
                'name'        => 'search',
                'field'       => ['todo.name'],
            ],
            'sort'          => [
                'latest'      => ['todo.time_stamp', _p('Latest')],
                'most-liked' => ['todo.total_like', _p('Most Liked')],
            ],
            'show'          => [10, 20, 30],
        ],
    ]);

    // Configure search service
    $aBrowseParams = [
        'module_id' => 'todo',
        'alias'     => 'todo',
        'field'     => 'task_id',
        'table'     => Phpfox::getT('todolist_task')
    ];

    $search->setContinueSearch(true);
    $search->browse()->params($aBrowseParams)->execute();
    $cnt = $search->browse()->getCount();
    $aItems = $search->browse()->getRows();

    // register pager service
    Phpfox_Pager::instance()->set([
        'page' => $search->getPage(),
        'size' => $search->getDisplay(),
        'count' => $search->browse()->getCount(),
    ]);
}

```

```

        // assign variables to template
        $this->template()->assign([
            'iCnt'      => $cnt,
            'aItems'    => $aItems,
            'sSearchBlock' => _p('search to do'),
            'bIsProfile' => $bIsProfile,
            'sTaskStatus' => $this->request()->get('status'),
            'sView'     => $sView,
        ]
    );
}
}
}

```

Edit **index.html.php**, paste following code.

```

{if !count($aItems)}
{if !PHPFOX_IS_AJAX}
<div class="extra_info">
    {_p var='no_to_to_list_item_found'}
</div>
{/if}
{else}
{if !PHPFOX_IS_AJAX}
<div class="item-collections item-collections-2">
{/if}

    {foreach from=$aItems name=aItem item=aItem}
    <article>
        <a href="{ permalink module='to-do-list.view' id=$aItem.task_id title=$aItem.name }"><h4>{$aItem.name}<
/h4></a>
        <p>{$aItem.description}</p>
        <abbr>
            <span>Status:</span><span>
                { if $aItem.task_status >0 }complete{else}in-complete{/if}
            </span><br/>
            <span>Created:</span><span>
                {$aItem.time_stamp|convert_time}
            </span><br/>
        </abbr>
    </article>
{/foreach}
{pager}
{if !PHPFOX_IS_AJAX}
</div>
{/if}
{/if}

```

Clear cache and visit to do list home page again, you see the tool bar at top. It is necessary to note that you need to clear cache whenever having changes in template files

 MEMBERS PHOTOS PAGES MUSIC EVENTS GROUPS QUIZZES POLLS FORUM VIDEOS ADVERTISE MARKETPLACE TO DO LIST

[Browse](#) [Create](#) 



To Do List

Latest ▼

10 per page ▼

All Time ▼

Doing homework

Working on upcoming assignment

Status: in-complete

Created: 1 second ago

Doing chores

Cleaning up room, working desk and making bed.

Status: in-complete

Created: 22 minutes ago

The view listing current user to do list using simple grid view, Each to do item has detail view located at `/to-do-list/view/:id/:name`, the next chapter will show you how to implement detail view of each task.

[Next Chapter](#)