

View To Do

View To-Do Item

Create PHP service class **ToDoList** under **Service** directory, past following code

```
<?php

namespace Apps\ToDoList\Service;

use Phpfox;

class ToDoList extends \Phpfox_Service
{
    public function getForBrowse($id)
    {
        return db()->select('t.*')
            ->from(Phpfox::getT('todolist_task'), 't')
            ->where('task_id=' . (int)$id)
            ->execute('getSlaveRow');
    }
}
```

Register this service to **start.php**

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

Create a new controller **ViewController** under **Controller** directory, then paste the following code

```

<?php

namespace Apps\ToDoList\Controller;

class ViewController extends \Phpfox_Component
{
    public function process(){

        $template = $this->template();

        // 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'));

        $id = $this->request()->get('req3');

        $browseService = \Phpfox::getService('todo');
        $aItem = $browseService->getForBrowse($id);

        $url = $this->url()->permalink('to-do-list.view', $aItem['task_id'], $aItem['name']);

        $template
            ->setTitle($aItem['name'])
            ->setBreadCrumb($aItem['name'], $url);

        $template->assign([
            'aItem'=>$aItem,
        ]);
    }
}

```

`$this->request()->get('req3')` is used to get the third value in the url. For example: [http://YourSite.com/index.php/req1/req2/req3/...](http://YourSite.com/index.php/req1/req2/req3/)

In this case, current url is: <http://YourSite.com/index.php/to-do-list/view/todo-id/todo-title>, so `request()->get('req3')` will return the id of the Todo List

Then, you have to register and add a route in **start.php** for this controller:

```

$module->addComponentNames('controller', [
    'todo.view' => Controller\ViewController::class,
]);

route('to-do-list/view/:id/:name', function () {
    \Phpfox_Module::instance()->dispatch('todo.view');
    return 'controller';
});

```

Let's review the previous chapter a little bit, after clicking the link

```

<a href="{ permalink module='to-do-list.view' id=$aItem.task_id title=$aItem.name }">

```

it will redirect you to the controller **todo.view** that you had registered in **start.php**, that means you will be redirected to the URL: <http://YourSite.com/index.php/to-do-list/view/todo-id/todo-title>, thus, the route for **todo.view** controller has to be 'to-do-list/view/:id/:name' in order to redirect and pass request parameters correctly.

Tip: You can also simply register route as: to-do-list/view/ - "*" means you can put as many requests as you want.*

After that, create a template file **view.html.php** under **views/controller** directory, paste following code

```
<h4>Name</h4>
<p>{$aItem.name}</p>
<h4>Description</h4>
<p>{$aItem.description}</p>
<h4>Status</h4>
<p>{ if $aItem.task_status >0 }complete{else}in-complete{/if}</p>
<h4>Created</h4>
<p>{$aItem.time_stamp|convert_time}</p>
```

*Why the file name is **view.html.php**, let's review the naming rule mentioned in the previous chapters.*

Integrate Comment To Page View

First, add code below to the bottom of file **view.html.php** under **views/controller**.

```
{ module name='feed.comment' }
```

By using { module name='...' }, you integrate block to your view.

Next, update function **getForBrowse** of **TodoList** service.

```
<?php
public function getForBrowse($id)
{
    if (Phpfox::isModule('friend')) {
        db()->select('f.friend_id AS is_friend, ')
            ->leftJoin(Phpfox::getT('friend'), 'f', "f.user_id = t.user_id AND f.friend_user_id = " . Phpfox::
getUserId());
    }

    if (Phpfox::isModule('like')) {
        db()->select('l.like_id AS is_liked, ')
            ->leftJoin(Phpfox::getT('like'), 'l', 'l.type_id = \'todo\' AND l.item_id = t.task_id AND l.user_id
= ' . Phpfox::getUserId());
    }

    return db()->select('t.*')
        ->from(Phpfox::getT('todolist_task'), 't')
        ->where('task_id=' . (int)$id)
        ->execute('getSlaveRow');
}
```

Finally, update **ViewController** with the following code.

```
<?php
namespace Apps\ToDoList\Controller;

class ViewController extends \Phpfox_Component
{
    public function process()
    {
        $template = $this->template();

        // 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'));

        $id = $this->request()->get('req3');

        $browseService = \Phpfox::getService('todo');
        $aItem = $browseService->getForBrowse($id);
        $url = $this->url()->permalink('to-do-list.view', $aItem['task_id'], $aItem['name']);

        $this->setParam('aFeed', array(
            'comment_type_id' => 'todo',
            'privacy' => $aItem['privacy'],
            'like_type_id' => 'todo',
            'feed_is_liked' => $aItem['is_liked'],
            'feed_is_friend' => $aItem['is_friend'],
            'item_id' => $aItem['task_id'],
            'user_id' => $aItem['user_id'],
            'total_comment' => $aItem['total_comment'],
            'total_like' => $aItem['total_like'],
            'feed_link' => \Phpfox_Url::instance()->makeUrl('to-do-list/view'). $aItem['task_id'],
            'feed_title' => $aItem['name'],
            'feed_display' => 'view',
            'feed_total_like' => $aItem['total_like'],
            'report_module' => 'todo',
            'report_phrase' => _p('report_this_todo'),
            'time_stamp' => $aItem['time_stamp']
        ));

        $template
            ->setTitle($aItem['name'])
            ->setBreadCrumb($aItem['name'], $url);

        $template->assign([
            'aItem' => $aItem,
        ]);
    }
}
```

Visit to do list homepage, click a detail title to be redirected to detailed page, the final result will be similar to



Doing homework

Name

Doing homework

Description

Working on upcoming assignment

Status

in-complete

Created

2 hours ago



Be the first person to like this. ...

AD Write a comment...



Congratulation, You have basic knowledge about Create, Listing, View a to-dopermalink list item, using phpFox services **database**, **URL**, **template**.

Next chapter will show you how to interact to most important of social network feature **Activity Feed**.

[Next Chapter](#)