

# Writing Your First App

## Requires

- *phpFox version >= 4.5.1*
- *Techie mode has already been enabled*

## What is app?

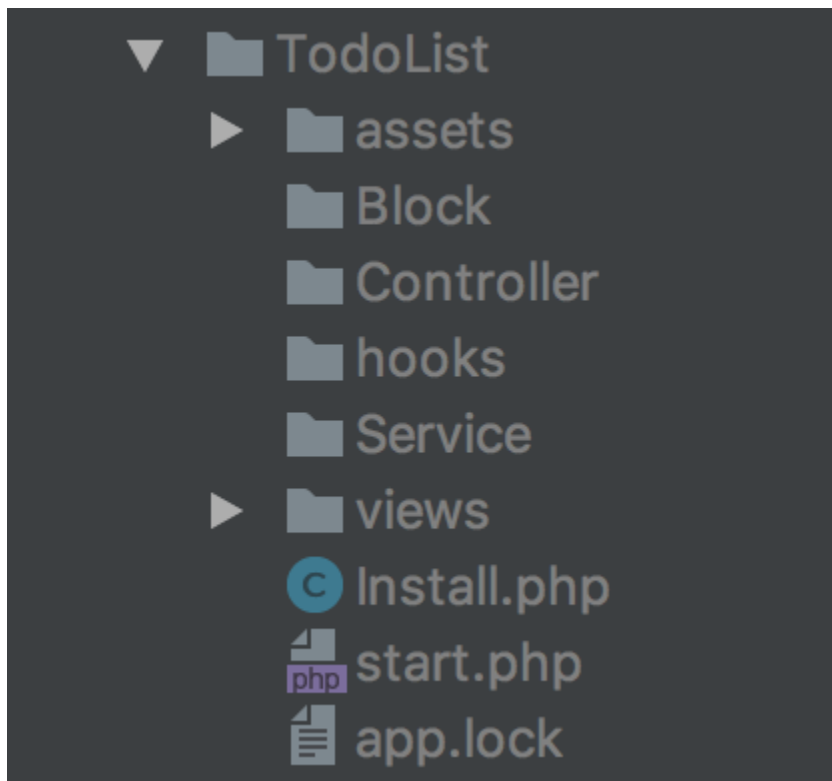
phpFox separates application structure to **App**. App provides new functions, gives new **blocks** that administrators put into template, integrate phpFox with external platforms etc Amazon S3, and even modify the way phpFox platform works.

## Creating a New App

The best way to get things started is to create a small app showing the most common API functions we use. Our example is creating a **ToDoList**, it's a simple **todolist** application, allow members share to do lists.

To create an app, go to AdminCP -> Apps -> Installed -> Actions dropdown -> New App -> put **YOUR\_APP\_ID** on the popup -> Click submit. Then check the folder `/PF.Site/Apps/YOUR_APP_ID/`, you will see some default files and folders there.

*APP\_ID is the name of your application*



Describe App Structure

- **Ajax:** This directory contains Ajax handler classes
- **assets:** This directory contains raw assets such as the images, css, javascript, ...
- **Block:** This directory contains **block** classes
- **Controller:** This directory contains **controller** classes
- **Service:** This directory contains **service** classes
- **hooks:** This directory contains **plugin** scripts
- **views:** This directory contain template scripts
- **phrase.json:** This file declares language phrases
- **icon.png:** This is icon of your app
- **Install.php:** This file contains installation script of your app
- **start.php:** This file contains bootstrap scripts.

*In some cases, you will have to modify permissions of these folder to be able to put your code in them.*

## Write Your First Controller

Add new **IndexController.php** file under directory **/PF.Site/Apps/ToDoList/Controller/** paste example code.

```
<?php

namespace Apps\ToDoList\Controller;

// Index controller must be 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

        // get url
        $url = $this->url()->makeUrl('to-do-list');

        $template->setBreadCrumb('To Do List',$url);

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

Add new template file **index.html.php** under **/PF.Site/Apps/ToDoList/views/controller,** paste following code

```
<p>Hello, to do list home page</p>
```

Now we define **route** to define configurations, edit **start.php**, paste following code.

```

<?php

namespace Apps\ToDoList;

// Load phpFox module service instance, this is core of phpFox service,
// module service contains your app configuration.
$module = \Phpfox_Module::instance();

// Instead of \Apps\FirstApp every where. Let register an alias **first_app** that map to our app.
$module->addAliasNames('todo', 'ToDoList');

// Register your controller here
$module->addComponentNames('controller', [
    'todo.index' => Controller\IndexController::class,
]);

// Register template directory
$module->addTemplateDirs([
    'todo' => PHPFOX_DIR_SITE_APPS . 'ToDoList/views',
]);

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

```

*All your php classes must have namespace **Apps\ToDoList**, This help the autoloader knows where to load scripts.*

*There is a rule of naming things: if you want to create a controller named **Index**, you should name the php file as **IndexController.php**; in **Start.php**, the route should be **book.index** the template file should be **index.html.php** (the first part of template file's name must be the same as the last part of the route - index)*

Open browse, in address bar append **/index.php/to-do-list/**, then see the result.

Browse Create

## To Do List

Hello, to do list home page

### Add Main Menu

Main menu will be added automatically by phpFox, edit Install.php,

update function setAlias

```
<?php
protected function setAlias()
{
    $this->alias = 'todo';
}
```

update function setOthers

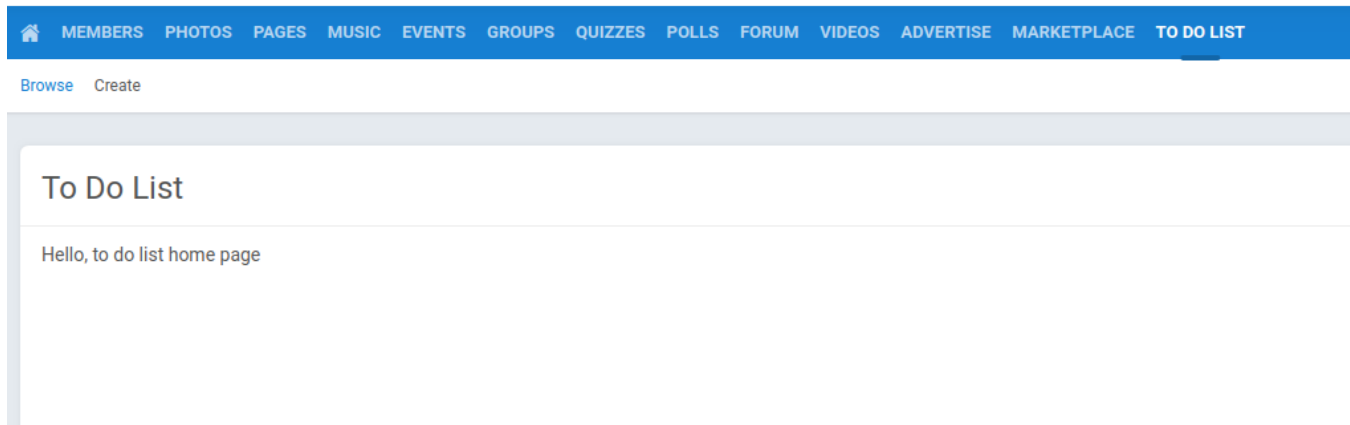
```
<?php
protected function setOthers()
{
    //add the menu for your app
    $this->menu = [
        "name" => "To Do List", // Menu label
        "url" => "/to-do-list", // Menu Url
        "icon" => "tasks" // Menu icons, see http://fontawesome.io/icons/
    ];
}
```

In case you want to use a defined phrase for menu label, you can use the below script:

```
<?php
$this->menu = [
    "phrase_var_name" => "menu_to_do_list", // Var name for the phrase of menu
    "url" => "/to-do-list", // Menu Url
    "icon" => "tasks" // Menu icons, see http://fontawesome.io/icons/
];
```

*Another way to use defined phrases is using **phrase.json** file, see detailed document [here](#)*

Then update your app to apply your modification. ([how to update app configurations](#))  
Open browser again, see result:



### Add Buttons

Buttons menu is handled by **Template** service, you can get template service from anywhere by using **Phpfox::getLib('template')**, within controller, service you can invoke method `template()` to get the Template service.

pasted code into `IndexController.php`

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



## To Do List

Hello, to do list home page

Click on **ADD TO DO**, a not found page will be shown, The next chapter will guide you how to create new controller to handle **Add** process, show input form about to do list task, store item to database.

[Next Chapter](#)