

Add New To Do

Add Controller

When a user clicks button **ADD TO DO**, it will redirects him to **Add To Do** page, which contains a form, he will fill information, then submits form to create his task, tasks will be shown into his to do list.

To do that, please follow below steps:

First of all, add new php class **AddController** under directory **Controller**, paste following code:

```
<?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(_p('Add To Do'));

        // set view breadcrumb
        $template->setBreadCrumb(_p('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'),
        ]);
    }
}
```

Look in to pasted code, you use `_p` function many times, what exactly `_p` does?

`_p` is a shortcut to call translation service, which translate a phrase id to the language which had been selected by current logged in user. `_p` gives a flexible, robust way allows developers/admins/translators to modify messages translation on the fly.

Example:

```
echo _p('Add To Do')
//english print: To do name is required
```

Translate method `_p()`

`_p` method is used to translate a message into multiple languages, the detail about translation will be explain later.

Add template file **add.html.php** under **views/controller**, paste following code

```

<form method="post">
  <div class="form-group">
    <label class="control-label">Name</label>
    <input type="text" name="val[name]" class="form-control"
      maxlength="255"/>
  </div>
  <div class="form-group">
    <label class="control-label">Description</label>
    <textarea name="val[description]" rows="3"
      class="form-control"></textarea>
  </div>
  <div class="form-group">
    <button type="submit" class="btn btn-primary" name="_submit">Submit</button>
  </div>
</form>

```

Register **controller** and **route** to **start.php**

```

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

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

```

Click on **+ Add To Do**, you see the result

The screenshot shows a web form titled "Add To Do". It contains two input fields: "Name" and "Description". Below the "Description" field is a blue "Submit" button. The form is styled with a light blue border and a white background.

At first we need a table to store to do list items.
 Create table **todolist_task** by running following query, by default phpFox uses **phpfox_** prefix for all tables,
 the final table name will be **phpfox_todolist_task**:

```
CREATE TABLE `phpfox_todolist_task` (  
  `task_id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `user_id` int(10) unsigned NOT NULL,  
  `name` varchar(255) NOT NULL DEFAULT '',  
  `description` varchar(255) NOT NULL DEFAULT '',  
  `task_status` tinyint(3) unsigned NOT NULL DEFAULT '0',  
  `time_stamp` int(10) unsigned NOT NULL,  
  `time_update` int(10) unsigned NOT NULL DEFAULT '0',  
  `privacy` tinyint(1) NOT NULL DEFAULT '0',  
  `total_comment` int(10) unsigned NOT NULL DEFAULT '0',  
  `total_view` int(10) unsigned NOT NULL DEFAULT '0',  
  `total_like` int(10) unsigned NOT NULL DEFAULT '0',  
  PRIMARY KEY (`task_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

- **user_id**: who creates task
- **name**: to do list name
- **description**: some lines about task
- **task_status**: status about task, 0: in-complete, 1: complete
- **time_update**: last modification time
- **time_stamp**: creation time
- **privacy**, **total_like**, **total_comment**, **privacy** will be explained later

Creating tables by sql is only for testing app, before exporting app package for publishing, you have to write script to create tables automatically when installing your app in another site. We will show you how to do that later.

After we have a table to store to do list task, continue editing AddController to handle process.

```

<?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
                \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
                ]);

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

```

Go to **+ Add To Do** page, full fill the form, then press submit, a new item will be added to database then user will be redirected to to do list home page. The next chapter will guide you how to show to do list items from database.

[Next Chapter](#)