

【一】基础说明

每一个应用插件都是一个独立的插件目录，所有的应用插件都是存放在appPlugin目录下。

以下示例是一个支付宝插登录插件的目录结构：

```
├─ AliLogin [插件名称]
│   └─ Services [服务目录，可选]
│       └─ Controller [控制器目录]
│           └─ BindController.class.php
│           └─ NotifyController.class.php
│               └─ View [视图目录，可选]
│                   └─ images
│                       └─ alipay.png
│                       └─ m_alipay.png
│                           └─ Mobile
│                               └─ dispatch_jump.html
│                               └─ login.html
│                                   └─ Pc
│                                       └─ dispatch_jump.html
│                                       └─ login.html
│                                           └─ config..html [配置文件页面]
├─ AliLoginPlugin.class.php [插件主文件]
├─ config.php [插件配置文件]
└─ README.md [插件说明文档，可选]
```

AliLoginPlugin.class.php文件为插件的入口文件，命名格式为：插件标识+Plugin.class.php

入口文件必须继承 Plugin.class.php 文件中的 Plugin 类；

```
<?php
namespace Plugin\AliLogin;
use Detection\MobileDetect;
use Plugin\Plugin;
class AliLoginPlugin extends Plugin
{
    /**
```

```

* 插件标识
* @var string
*/
public $code = 'AliLogin';
/**
* @var string 自定义设置模板
*/
public $custom_config = 'View/config.html';
/**
* 插件信息
* @var array
*/
public $info
    = array(
        'name'          => '支付宝',
        'title'         => '支付宝快捷登录',
        'description'   => '支付宝快捷登录',
    );
/**
* 插件挂载钩子
* @var array
*/
public $hooks = ['addLogin', 'loginAuth', 'loginPluginCode'];
/**
* 安装程序
* @return bool|mixed
*/
public function install()
{
    S('hooks', null);
    return true;
}
/**
* 卸载程序
* @return bool|mixed
*/
public function uninstall()
{
    return true;
}
/**
* 登录方式加载
*
* @param array $param
*/
public function addLogin(&$param = [])
{
    if ($this->verify()) {
        $mobile = new MobileDetect();
        /** 检测是否是手机网站 **/
        $isMobile = $mobile->isMobile();
        $url      = SITE_DOMAIN .

```

```

"/ApiNotify/Notify/url/plugin/AliLogin";
    $this->assign('auth_url', $url);
    if ($isMobile) {
        $this->display('View/Mobile/login');
    } else {
        $this->display('View/Pc/login');
    }
}
}
public function loginAuth(&$param = [])
{
    if ($this->verify()) {
        $url = SITE_DOMAIN . "/ApiNotify/Bind/url/plugin/AliLogin";
        $this->assign('auth_url', $url);
        $mobile = new MobileDetect();
        if ($mobile->isMobile()) {
            $tips = '支付宝绑定与解绑操作，跳转支付宝后，需要重新登陆一次，敬请注意。';
        } else {
            $tips = '';
        }
        $param[$this->code] = [
            'url' => $url, //授权跳转的URL
            'name' => $this->info['name'], //插件名，在用户中心第三方登录列表展
            'code' => $this->code, //插件识别码
            'tips' => $tips, //向用户展示的插件提示语，没有可返回空
        ];
    }
}
public function loginPluginCode(&$params)
{
    if ($this->verify()) {
        $params[] = $this->code;
    }
}
}

```

示

【二】关于3个钩子方法的说明

- PluginAliLoginAliLoginPlugin::addLogin()

template/Home/Zkeys/PC/Page/Login/login.html

和template/Home/Zkeys/Mobile/Page/Login/login.html通过钩子把页面嵌入到登录页内

```
<!-- 登录插件start -->
```

```
{:hook('addLogin')}
```

```
<!-- 登录插件end -->
```

- PC端 app/Plugin/AliLogin/View/Pc/login.html

```
<style>
    .alipay-ico{
        display: inline-block;
        height: 24px;
        width: 24px;
        border-radius: 50%;
        background: url('/app/Plugin/AliLogin/View/images/alipay.png') no-
repeat center;//第三方登录图标, 建议80px*80px
        background-size: cover;
        margin-right:10px;
    }
</style>
<a class="alipay-ico" href="{ $auth_url}" target="_blank"></a>
```

- 移动端 app/Plugin/AliLogin/View/Mobile/login.html

```
<style>
    .alipay-ico{
        display: inline-block;
        height: 1.4rem;
        width: 1.4rem;
        border-radius: 50%;
        background: url('/app/Plugin/AliLogin/View/images/m_alipay.png') no-
repeat center;//第三方登录图标, 建议24px*24px
        background-size: cover;
        margin-right:1.5rem;
    }
</style>
<a class="alipay-ico plugin-btn" href="{ $auth_url}" target="_blank"></a>
```

其中 \$auth_url 是跳转到登录扫码授权页, 这里是先跳转到 http(s)://域名/ApiNotify/Notify/url/plugin/AliLogin 实际访问的是 app/Plugin/AliLogin/Controller/NotifyController.class.php::url() 然后重定向到登录扫码授权页

假设授权后回调地址为 http(s)://域名/ApiNotify/Notify/login/plugin/AliLogin[] 则处理伪代码如下:

```
$where = [
    'type' => 'AliLogin',
    'auth_user' => $aliUserId,
];
$authLogin = M('auth_login')->where($where)->find();
if (empty($authLogin)) {
    $this->error('账户没有被绑定, 请登录并在 <strong>账号设置</strong>第三方账号绑定</strong> 绑定后重试', "/login");
    return;
}
$userinfo = M('users')->where(['id' => $authLogin['user_id'], 'status' => 1])->find();
if (empty($userinfo)) {
```

```

        $this->error('此支付宝绑定的用户状态异常或不存在', "/login");
        return;
    }
    $usersObj = new \Niaoyun\User\Users();
    $result = $usersObj->loginSuccess($userinfo, 5);
    if (\Niaoyun\User\Users::IDENTITY_STATUS_YES == $userinfo['identitystatus'])
    {
        $userinfo['name'] = encryptAuthName($userinfo['name']);
    }
    $notice = $userinfo["name"] ? $userinfo["name"] :
    encryptionMobile($userinfo['mobile']);
    if ($result['result']) {
        $this->assign('customSecond', 0);
        $this->success('亲爱的' . $notice . '欢迎回到用户中心', '/user');
    } else {
        $this->error($result['text'], '/login');
    }
}

```

▪ PluginAliLoginAliLoginPlugin::loginAuth()

其中 \$auth_url 是跳转到登录扫码授权页，这里是先跳转到 http(s)://域名/ApiNotify/Bind/url/plugin/AliLogin 实际访问的
是app/Plugin/AliLogin/Controller/BindController.class.php::url()然后重定向到登录扫码授权页

假设授权后回调地址为http(s)://域名/ApiNotify/Bind/index/plugin/AliLogin则处理伪代码如下

//处理后的跳转地址

```

        $url = '/user/oauth/index';
        $authLoginModel = M('auth_login');
        $where['_string'] = "(user_id = '{$this->userinfo['id']}' or
auth_user = '{$aliUserId}') and type = 'AliLogin'";
        $authLogin = $authLoginModel->where($where)->find();
        if ($authLogin) {
            $this->assign('customSecond', 3);
            $this->error('您已经绑定了支付宝账号或者该支付宝账号绑定了其他用户',
$url);
        }
        $add = [
            'user_id' => $this->userinfo['id'],
            'auth_user' => $aliUserId,
            'type' => 'AliLogin',
            'create_time' => date('Y-m-d H:i:s'),
            'other' => '',
        ];
        $id = $authLoginModel->add($add);
        if ($id === false) {
            $this->error('绑定失败', $url);
            return;
        }
    }
}

```

```
$this->success('绑定成功', $url);
```

- PluginAliLoginAliLoginPlugin::loginPluginCode() 返回当前插件标识即可。

其他

\$this->error()或\$this->success()需要自行添加跳转模板。

PC的复制 template/Home/Zkeys/PC/Page/dispatch_jump.html

移动端的复制 template/Home/Zkeys/Mobile/Page/dispatch_jump.html

app/Plugin/AliLogin/Controller/NotifyController.class.php

```
class NotifyController extends \Common\Controller\HomeController
{
    public function __construct()
    {
        parent::__construct();
        $mobile = new \Detection\MobileDetect();
        /** 检测是否是手机网站 */
        if ($mobile->isMobile()) {
            C('TMPL_ACTION_ERROR', REAL_PATH .
'/app/Plugin/AliLogin/View/Mobile/dispatch_jump.html');
            C('TMPL_ACTION_SUCCESS', REAL_PATH .
'/app/Plugin/AliLogin/View/Mobile/dispatch_jump.html');
        } else {
            C('TMPL_ACTION_ERROR', REAL_PATH .
'/app/Plugin/AliLogin/View/Pc/dispatch_jump.html');
            C('TMPL_ACTION_SUCCESS', REAL_PATH .
'/app/Plugin/AliLogin/View/Pc/dispatch_jump.html');
        }
    }
    public function login(){
    public function url(){
}
}
app/Plugin/AliLogin/Controller/BindController.class.php
```

app/Plugin/AliLogin/Controller/BindController.class.php

```
class BindController extends \Common\Controller\UserController
{
    public function __construct()
    {
        $mobile = new \Detection\MobileDetect();
        /** 检测是否是手机网站 */
        if ($mobile->isMobile()) {
            C('TMPL_ACTION_ERROR', REAL_PATH .
'/app/Plugin/AliLogin/View/Mobile/dispatch_jump.html');
            C('TMPL_ACTION_SUCCESS', REAL_PATH .
'/app/Plugin/AliLogin/View/Mobile/dispatch_jump.html');
        } else {
            C('TMPL_ACTION_ERROR', REAL_PATH .
'/app/Plugin/AliLogin/View/Pc/dispatch_jump.html');
        }
    }
}
app/Plugin/AliLogin/Controller/BindController.class.php
```

```
                C('TMPL_ACTION_SUCCESS', REAL_PATH .  
'/app/Plugin/AliLogin/View/Pc/dispatch_jump.html');  
            }  
            parent::__construct();  
        }  
        public function index(){  
        public function url(){  
    }  
}
```