

20220415 评估

相关链接

- [fastadmin官方文档](#)
- [thinkphp5官方文档](#)
- [think-template模板引擎官方文档](#)
- [adminLTE前端框架官方文档](#)
- [Bootstrap3.4官方文档](#)

安装

1、下载安装包 [完整包](#) 必须要使用完整包

2、安装环境要求

- PHP >= 7.1 且 7.3 (推荐PHP7.1版本)
- Mysql >= 5.6 (需支持innodb引擎)
- Apache 或 Nginx
- PDO PHP Extension
- MBstring PHP Extension
- CURL PHP Extension
- Git (用于源代码的版本管理)
- Node.js (可选,用于安装Bower和LESS,同时打包压缩也需要使用到)
- Composer (可选,用于管理第三方扩展包)
- Bower (可选,用于管理前端资源)
- Less (可选,用于编辑less文件,如果你需要增改css样式,建议安装上)
- 宝塔伪静态请配置使用thinkphp规则

lamp\conf\httpd.conf配置 LoadModule rewrite_module modules/mod_rewrite.so

入口文件夹增加.htaccess文件

```
<IfModule mod_rewrite.c>
  RewriteEngine on
  RewriteBase /
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteCond %{REQUEST_FILENAME} !-f
  RewriteRule ^(.*)$ index.php?s=/$1 [QSA,PT,L]
```

</IfModule>

3、运行网站执行安装

4、常见问题

- 如果安装后打开页面提示404错误，请检查服务器伪静态配置，宝塔伪静态请配置使用thinkphp规则。
- 提示请先下载完整包覆盖后再安装，说明你是直接从仓库下载的代码，请从官网下载完整包覆盖后再进行安装。
- 如果提示当前权限不足，无法写入配置文件application/database.php请检查database.php是否可读，还有可能是当前安装程序无法访问父目录，请检查PHP的open_basedir配置。
- 如果提示找不到fa_admin表或表不存在，请检查你的MySQL是否开启了支持innodb
- 如果使用的宝塔面板，请在软件配置中PHP的禁用函数中，移除putenv函数。

框架浅析

1、支持命令行创建基本代码，类似Yii[具体命令看这里](#)

2、后端部分：

- 框架基于ThinkPHP5.0
- 模板引擎使用的是think-template
- 使用模块化的开发方式
- 支持插件开发，比如：[在线命令插件](#)
- 权限校验，使用AUTH模式，区分后台管理权限与前台会员权限，模板中校验权限{:\$auth→check('article/vip')}]auth模块/application/admin/controller/auth
- 支持FormBuilder工具,样式使用的是bootstrap3如：{:Form::text('row[text]', '[data-rule⇒'required'])}]具体代码文件/extend/fast/Form.php

```
//生成Token
Form::token()
//Label标签
Form::label(string $name, string $value = null, array $options = [])
//按类型生成文本框
Form::input($type, $name, string $value = null, array $options = [])
//普通文本框
Form::text(string $name, string $value = null, array $options = [])
//密码文本框
Form::password(string $name, array $options = [])
//隐藏文本框
Form::hidden(string $name, string $value = null, array $options = [])
//Email文本框
Form::email(string $name, string $value = null, array $options = [])
//URL文本框
Form::url(string $name, string $value = null, array $options = [])
//文件组件
Form::file(string $name, array $options = [])
//多行文本框
Form::textarea(string $name, string $value = null, array $options = [])
```

```
//富文本编辑器
Form::editor(string $name, string $value = null, array $options = [])
//下拉列表组件
Form::select(string $name, array $list = [], string $selected = null, array $options = [])
//下拉列表组件(多选)
Form::selects(string $name, array $list = [], string $selected = null, array $options = [])
//下拉列表组件(友好)
Form::selectpicker(string $name, array $list = [], string $selected = null, array $options = [])
//下拉列表组件(友好)(多选)
Form::selectpickers(string $name, array $list = [], string $selected = null, array $options = [])
//动态下拉列表组件
Form::selectpage(string $name, string $value, string $url, string $field = null, string $primaryKey = null, array $options = [])
//动态下拉列表组件(多选)
Form::selectpages(string $name, string $value, string $url, string $field = null, string $primaryKey = null, array $options = [])
//城市选择组件
Form::citypicker(string $name, string $value, array $options = [])
//开关组件
Form::switcher(string $name, string $value, array $options = [])
//日期选择组件
Form::datepicker(string $name, string $value, array $options = [])
//时间选择组件
Form::timepicker(string $name, string $value, array $options = [])
//日期时间选择组件
Form::datetimepicker(string $name, string $value, array $options = [])
//日期区间组件
Form::daterange(string $name, string $value, array $options = [])
//时间区间组件
Form::timerange(string $name, string $value, array $options = [])
//日期时间区间组件
Form::datetimerange(string $name, string $value, array $options = [])
//字段列表组件
Form::fieldlist(string $name, string $value, string $title = null, string $template = null, array $options = [])
//联动组件
Form::cxselect(string $url, array $names = [], array $values = [], array $options = [])
//选择数字区间
Form::selectRange(string $name, string $begin, string $end, string $selected = null, array $options = [])
//选择年
Form::selectYear(string $name, string $begin, string $end, string $selected = null, array $options = [])
//选择月
Form::selectMonth(string $name, string $selected = null, array $options = [], string $format = '%m')
//单个复选框
Form::checkbox(string $name, string $value = '1', string $checked = null, array $options = [])
```

```
//一组复选框
Form::checkboxs(string $name, array $list = [], string $checked = null,
array $options = [])
//单个单选框
Form::radio(string $name, string $value = null, string $checked = null,
array $options = [])
//一组单选框
Form::radios(string $name, array $list = [], string $checked = null, array
$options = [])
//上传图片组件
Form::image(string $name = null, string $value, array $inputAttr = [], array
$uploadAttr = [], array $chooseAttr = [], array $previewAttr = [])
//上传图片组件(多图)
Form::images(string $name = null, string $value, array $inputAttr = [],
array $uploadAttr = [], array $chooseAttr = [], array $previewAttr = [])
//上传文件组件
Form::upload(string $name = null, string $value, array $inputAttr = [],
array $uploadAttr = [], array $chooseAttr = [], array $previewAttr = [])
//上传文件组件(多文件)
Form::uploads(string $name = null, string $value, array $inputAttr = [],
array $uploadAttr = [], array $chooseAttr = [], array $previewAttr = [])
//表单button
Form::button(string $value = null, array $options = [])
```

* 文件结构

FastAdmin项目目录

```
|— addons //插件存放目录
|— application //应用目录
|   |— admin //后台管理应用模块
|   |— api //API应用模块
|   |— common //通用应用模块
|   |— extra //扩展配置目录
|   |— index //前台应用模块
|   |— build.php
|   |— command.php //命令行配置
|   |— common.php //通用辅助函数
|   |— config.php //基础配置
|   |— database.php //数据库配置
|   |— route.php //路由配置
|   |— tags.php //行为配置
|— extend
|   |— fast //FastAdmin扩展辅助类目录
|— public
|   |— assets
|   |   |— build //打包JS/CSS的资源目录
|   |   |— css //CSS样式目录
|   |   |— fonts //字体目录
|   |   |— img
|   |   |— js
```

```

├── backend
├── frontend //后台功能模块JS文件存放目录
├── libs //Bower资源包位置
├── less //Less资源目录
├── uploads //上传文件目录
├── index.php //应用入口主文件
├── install.php //FastAdmin安装引导
├── admin.php //后台入口文件(自动安装后会被修改为随机文件名)
├── robots.txt
├── router.php
├── runtime //缓存目录
├── thinkphp //ThinkPHP5框架核心目录
├── vendor //Composer资源包位置
├── .bowerrc //Bower目录配置文件
├── .env.sample //环境配置模板(可复制一份为 .env 生效)
├── LICENSE
├── README.md
├── bower.json //Bower前端包配置
├── build.php
├── composer.json //Composer包配置
├── think //命令行控制台入口(使用 php think 命令进入)

```

- 一个模块具体文件结构

```

├── application
│   ├── admin
│   │   ├── controller
│   │   │   └── Test.php //控制器类
│   │   ├── lang
│   │   │   ├── zh-cn
│   │   │   │   └── test.php //功能语言包,按需加载
│   │   │   └── zh-cn.php //后台语言包,默认加载
│   │   ├── model
│   │   │   └── Test.php //模型类
│   │   ├── validate
│   │   │   └── Test.php //验证器类
│   │   └── view
│   │       └── test
│   │           ├── index.html //列表视图
│   │           ├── add.html //添加视图
│   │           └── edit.html //编辑视图
├── public
│   ├── assets
│   │   └── js
│   │       └── backend
│   │           └── test.js //功能模块JS文件

```

- 权限控制

//控制器中两个属性设置每个方法权限过滤

protected \$noNeedLogin = []; //无需登录的方法,同时也就不需要鉴权了

```
protected $noNeedRight = []; //无需鉴权的方法,但需要登录
```

具体权限配置，再后台中设置

fast-admin二开

Admin 在线

搜索菜单

控制台中

系统管理

权限管理

管理用户

管理用户日志

角色组

菜单规则

操作管理

会员管理

test

菜单规则

菜单规则通常对应一个控制器的方法,同时菜单数据也从规则中获取

新增

编辑

删除

更多

显示全部

ID	标题	图标	规则	权重	状态	菜单	操作
1	控制台	dashboard		143	正常		
2	系统管理	general		137	正常		
6	系统配置	generalconfig		60	正常		
7	附件管理	generalattachment		53	正常		
8	个人资料	generalprofile		34	正常		
3	分类管理	category		119	正常		
5	权限管理	auth		99	正常		
9	管理用户	auth/admin		118	正常		
10	管理用户日志	auth/adminlog		113	正常		
11	角色组	auth/group		109	正常		
12	菜单规则	auth/hule		104	正常		

fast-admin二开

Admin 在线

搜索菜单

控制台中

系统管理

权限管理

管理用户

管理用户日志

角色组

菜单规则

操作管理

会员管理

test

会员规则

会员规则通常对应一个控制器的方法,同时会员数据也从规则中获取

新增

编辑

删除

更多

显示全部

ID	标题	规则	备注	是否菜单	权重	状态	操作
2	API接口	api			2	正常	
4	会员模块	user			11	正常	
11	会员中心	api/user/index			10	正常	
10	注册	api/user/register			8	正常	
9	登录	api/user/login			6	正常	
12	个人资料	api/user/profile			3	正常	
1	前台	index			1	正常	
16	文章模块	article			16	正常	
18	文章vip	index/article/vip			18	正常	
17	文章中心	index/article/index			17	正常	
3	会员模块	user			12	正常	
7	会员中心	index/user/index			9	正常	

- Api的开发按照thinkphp5.0的方式开发即可，只是多了权限控制功能，如application/api/controller/User.php

```
class User extends Api
{
    protected $noNeedLogin = ['login', 'mobilelogin', 'register', 'resetpwd', 'changeemail', 'changemobile', 'third'];
    protected $noNeedRight = '*';

    public function _initialize()
    { ...
    }

    /**
     * 会员中心
     */
    public function index()
    { ...
    }

    /**
     * 会员登录
     *
     * @ApiMethod (POST)
     * @param string $account 账号
     * @param string $password 密码
     */
    public function login()
    {
        $account = $this->request->post('account');
        $password = $this->request->post('password');
        if (!$account || !$password) {
            $this->error(__('Invalid parameters'));
        }
        $ret = $this->auth->login($account, $password);
        if ($ret) {
            $data = ['userinfo' => $this->auth->getUserinfo()];
            $this->success(__('Logged in successful'), $data);
        } else {
            $this->error($this->auth->getError());
        }
    }
}
```

POST 192.168.100.80:82/api/user/login

Header Query **Body** 认证 预执行脚本 后执行脚本

< Body参数 ↑ 导入参数 ↓ 导出参数

<input checked="" type="checkbox"/>	account	Text ▾	Brightness	必
<input checked="" type="checkbox"/>	password	Text ▾	123456	必

实时响应 请求头 (8) 响应头 (11) Cookie (0) 成功响应示例及参数描述 错误响应示例及参数描述

美化 原生 预览 断言 🔊 🔍 绑定响应结果到变量?

```

1 {
2   "code": 1,
3   "msg": "登录成功",
4   "time": "1649902029",
5   "data": {
6     "userinfo": {
7       "id": 2,
8       "username": "Brightness",
9       "nickname": "Brightness",
10      "mobile": "13888888889",
11      "avatar": "
          0iMTAwIj48L3JlY3Q+PGRleHQgeD0iNTAiIHk9IjUwIi8mb250LXNpemU9IjUwIi80ZXh0LWNvbk9mZhc3QiIGZpbGw9IiNmZmZmZW
12      ",
13       "score": 0,
14       "token": "024754ca-2863-4f5f-8ed4-eafc7fa65d7b",
15       "user_id": 2,
16       "createtime": 1649902029,
17       "expiretime": 1652494029,
18       "expires_in": 2592000
19     }
20   }
  
```

3、前端部分：

- 基于AdminLTE 和 Bootstrap
- Requestjs处理js模块化

```
define(['jquery', 'bootstrap', 'backend', 'addtabs', 'adminlte', 'form'],  
function ($, undefined, Backend, undefined, AdminLTE, Form) {  
    var Controller = {  
        index: function () {  
            //to do ...  
        },  
    };  
  
    return Controller;  
});
```

demo

1、创建数据表，每个表以及字段都要填写注释，用于命令生成模型别名和字段label[]必须使用innoDB

```
DROP TABLE IF EXISTS `fa_article`;  
CREATE TABLE `fa_article` (  
    `id` INT(11) NOT NULL AUTO_INCREMENT COMMENT 'id',  
    `title` VARCHAR(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT  
NULL COMMENT '标题',  
    `article_image` VARCHAR(255) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NULL DEFAULT NULL COMMENT '图片',  
    `article_content` VARCHAR(255) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_unicode_ci NOT NULL COMMENT '内容',  
    `articlecatcat_id` INT(11) NULL DEFAULT 0 COMMENT '分类',  
    PRIMARY KEY (`id`) USING BTREE  
) ENGINE = InnoDB AUTO_INCREMENT = 4 CHARACTER SET = utf8mb4 COLLATE =  
utf8mb4_unicode_ci ROW_FORMAT = Dynamic;  
  
SET FOREIGN_KEY_CHECKS = 1;  
  
DROP TABLE IF EXISTS `fa_articlecat`;  
CREATE TABLE `fa_articlecat` (  
    `id` INT(11) NOT NULL AUTO_INCREMENT COMMENT 'id',  
    `name` VARCHAR(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT  
NULL COMMENT '分类名称',  
    PRIMARY KEY (`id`) USING BTREE  
) ENGINE = InnoDB AUTO_INCREMENT = 3 CHARACTER SET = utf8mb4 COLLATE =  
utf8mb4_unicode_ci COMMENT = '文章分类' ROW_FORMAT = Dynamic;  
  
SET FOREIGN_KEY_CHECKS = 1;
```

2、执行命令，生成基本代码


```
//创建分类crud
$ php think crud --table=fa_articlecat --controller=article/category
//创建文章crud
$ php think crud --table=fa_article --controller=article/article --
relation=fa_articlecat --relationmode=belongsto --
relationforeignkey=articlecatcat_id --relationprimarykey=id --
relationfields=name
//创建分类菜单
$ php think menu --controller=article/Category
//创建文章菜单
$ php think menu --controller=article/Article
```

From:

<http://192.168.100.64:58080/> - 昊维知识库

Permanent link:

http://192.168.100.64:58080/doku.php?id=hoeron:home:soft_dev:fastadmin:eval_20220414

Last update: **2022/04/18 18:16**

