[PHP]\_TP5.0\_FastAdmin\_命令行；

--王尔贝(我是害虫)

(整理 <https://doc.fastadmin.net/docs/command.html>)

--

FastAdmin基于ThinkPHP5强大的命令行功能扩展了一系列命令行功能，可以很方便的一键生成CRUD、生成权限菜单、压缩打包CSS和JS、安装配置插件等功能。

--

**一、一键生成CRUD**

在FastAdmin中可以快速的一键生成CRUD，其中包括控制器、模型、视图、验证器、语言包、JS。

//

**1、准备工作**

在数据库中创建一个fa\_test数据表，编辑好表字段结构，并且一定写上字段注释和表注释，相关数据表字段的设计要求可以参考数据库章节。FastAdmin在生成CRUD时会根据字段属性、字段注释、表注释自动生成语言包、组件和排版。请确保php所在的目录已经加入到系统环境变量，否则会提示找不到该命令。打开命令行控制台进入到FastAdmin根目录，也就是think文件所在的目录。

//

**2、常用命令**

// 生成fa\_test表的CRUD

php think crud -t test

// 生成fa\_test表的CRUD且一键生成菜单

php think crud -t test -u 1

// 删除fa\_test表生成的CRUD

php think crud -t test -d 1

// 生成fa\_test表的CRUD且控制器生成在二级目录下

php think crud -t test -c mydir/test

// 生成fa\_test\_log表的CRUD且生成对应的控制器为testlog

php think crud -t test\_log -c testlog

// 生成fa\_test表的CRUD且对应的模型名为testmodel

php think crud -t test -m testmodel

// 生成fa\_test表的CRUD且生成关联模型category，外链为category\_id，关联表主键为id

php think crud -t test -r category -k category\_id -p id

// 生成fa\_test表的CRUD且所有以list或data结尾的字段都生成复选框

php think crud -t test --setcheckboxsuffix=list --setcheckboxsuffix=data

// 生成fa\_test表的CRUD且所有以image和img结尾的字段都生成图片上传组件

php think crud -t test --imagefield=image --imagefield=img

// 关联多个表,参数传递时请按顺序依次传递，支持以下几个参数relation/relationmodel/relationforeignkey/relationprimarykey/relationfields/relationmode

php think crud -t test --relation=category --relation=admin --relationforeignkey=category\_id --relationforeignkey=admin\_id

// 生成v\_phealth\_db2数据库下的fa\_test表的CRUD

php think crud -t test --db=v\_phealth\_db2

//

**3、参数介绍**

-t, --table=TABLE 表名，带不带表前缀均可

-c, --controller[=CONTROLLER] 生成的控制器名,可选,默认根据表名进行自动解析

-m, --model[=MODEL] 生成的模型名,可选,默认根据表名进行自动解析

-i, --fields[=FIELDS] 生成的数据列表中可见的字段，默认是全部

-f, --force[=FORCE] 是否覆盖模式,如果目标位置已经有对应的控制器或模型会提示

-l, --local[=LOCAL] 是否本地模型,默认1,置为0时,模型将生成在common模块下

-r, --relation[=RELATION] 关联模型表名，带不带表前缀均可

-e, --relationmodel[=RELATIONMODEL] 生成的关联模型名,可选,默认根据表名进行自动解析

-k, --relationforeignkey[=RELATIONFOREIGNKEY] 表外键,可选,默认会识别为使用 模型\_id 名称

-p, --relationprimarykey[=RELATIONPRIMARYKEY] 关联模型表主键,可选,默认会自动识别

-s, --relationfields[=RELATIONFIELDS] 关联模型表显示的字段，默认是全部

-o, --relationmode[=RELATIONMODE] 关联模型,hasone或belongsto [default: "belongsto"]

-d, --delete[=DELETE] 删除模式,将删除之前使用CRUD命令生成的相关文件

-u, --menu[=MENU] 菜单模式,生成CRUD后将继续一键生成菜单

--db[=key] 多数据库支持(参数为tp5中配置的数据库key 在application\config.php添加数据库配置信息)

--setcheckboxsuffix[=SETCHECKBOXSUFFIX] 自动生成复选框的字段后缀

--enumradiosuffix[=ENUMRADIOSUFFIX] 自动生成单选框的字段后缀

--imagefield[=IMAGEFIELD] 自动生成图片上传组件的字段后缀

--filefield[=FILEFIELD] 自动生成文件上传组件的字段后缀

--intdatesuffix[=INTDATESUFFIX] 自动生成日期组件的字段后缀

--switchsuffix[=SWITCHSUFFIX] 自动生成可选组件的字段后缀

--citysuffix[=CITYSUFFIX] 自动生成城市选择组件的字段后缀

--selectpagesuffix[=SELECTPAGESUFFIX] 自动生成Selectpage组件的字段后缀

--ignorefields[=IGNOREFIELDS] 排除的字段

--editorclass[=EDITORCLASS] 自动生成富文本组件的字段后缀

--headingfilterfield[=HEADINGFILTERFIELD] 自动生成筛选过滤选项卡的字段，默认是status字段

--sortfield[=SORTFIELD] 排序字段

//

**4、常见问题**

(1)、如果你的表带有下划级会自动生成带层级的控制器和视图，如果你不希望生成带层级的控制器和视图，请使用-c 参数，例如：php think crud -t test\_log -c testlog将会生成testlog这个控制器，同理如果你的普通表想生成带层级的控制器则可以使用php think crud -t test -c mydir/test。

(2)、FastAdmin自带一个fa\_test表用于测试CRUD能支持的字段名称和类型，请直接使用php think crud -t test生成查看。

(3)、生成CRUD后，关联表外键在列表未显示对应的关联表数据信息，此时建议你使用在线命令行插件进行可视化生成。

(4)、生成CRUD后，在添加或编辑时外键字段未能正确显示关联表数据列表，请查看数据库章节常见问题中的说明。

(5)、如果需要生成回收站，请务必保证你的表中存在deletetime字段，且默认值为null，这样在一键CRUD时将自动生成回收站的功能，如果deletetime默认值为0，数据将默认进入回收站。

//

**5、使用范例**

更多CRUD一键生成可使用的参数请使用php think crud --help查看。

--

**二、一键生成菜单**

FastAdmin可通过命令控制台快速的一键生成后台的权限节点，同时后台的管理菜单也会同步改变，操作非常简单。

//

**1、准备工作**

首先确保已经将FastAdmin配置好，数据库连接正确，同时确保已经通过上一步的一键生成CRUD已经生成了test的CRUD。请确保php所在的目录已经加入到系统环境变量，否则会提示找不到该命令。打开控制台进入到FastAdmin根目录，也就是think文件所在的目录。

//

**2、常用命令**

// 一键生成test控制器的权限菜单

php think menu -c test

// 一键生成mydir/test控制器的权限菜单

php think menu -c mydir/test

// 删除test控制器生成的菜单

php think menu -c test -d 1

// 一键全部重新所有控制器的权限菜单

php think menu -c all-controller

//

**3、常见问题**

(1)、在使用php think menu前确保你的控制器已经添加或通过php think crud生成好。

(2)、如果之前已经生成了菜单,需要再次生成,请登录后台手动删除之前生成的菜单或使用php think menu -c 控制器名 -d 1来删除。

(3)、如果生成层级目录的菜单，在后台展示时父级菜单会以目录名称显示，如果需要修改可以在application/admin/lang/zh-cn.php中追加相应的语言包即可。

//

**4、使用范例**

更多CRUD一键生成可使用的参数请使用php think menu --help查看。

--

**三、一键压缩打包**

在FastAdmin中如果修改了核心的JS或CSS文件，是需要重新压缩打包后在生产环境下才会生效。FastAdmin采用的是基于RequireJS的r.js进行JS和CSS文件的压缩打包。

//

**1、准备工作**

请先确保你的环境已经安装好Node环境。首先确认你application/config.php中app\_debug的值，当为true的时候是采用的无压缩的JS和CSS，当为false时采用的是压缩版的JS和CSS。请确保php所在的目录已经加入到系统环境变量，否则会提示找不到该命令。打开命令行控制台进入到FastAdmin根目录，也就是think文件所在的目录。

//

**2、常用命令**

// 一键压缩打包前后台的JS和CSS

php think min -m all -r all

// 一键压缩打包后台的JS和CSS

php think min -m backend -r all

// 一键压缩打包前后台的JS

php think min -m all -r js

// 一键压缩打包后台的CSS

php think min -m backend -r css

//

**3、常见问题**

Windows系统需要手动配置node的路径,请参考在Windows下如何压缩打包JS和CSS。如果无法进行打包，可以使用php think min -m all -r all -vvv尝试下，看下错误信息。如果压缩打包后访问不生效，请检查是否是你的浏览器缓存的原因。请不要直接修改以.min.js和.min.css结尾的文件，因为一键压缩打包后会进行覆盖。安装或卸载插件后无需进行压缩打包JS和CSS。

//

**4、影响文件**

在调试模式和生产环境下所加载的JS和CSS是不一样的，压缩打包会重新生成相应环境下的JS和CSS文件，特别注意下所述。

// 调试模式

public/assets/js/require-frontend.js

public/assets/js/require-backend.js

public/assets/css/frontend.css

public/assets/css/backend.css

// 生产环境(打包压缩后会重新生成以下文件)

public/assets/js/require-frontend.min.js

public/assets/js/require-backend.min.js

public/assets/css/frontend.min.css

public/assets/css/backend.min.css

//

**5、使用范例**

JS和CSS文件压缩前和压缩后浏览器请求对比(请右键查看大图)。更多一键生成JS和CSS的参数请使用php think min --help查看。

--

**四、一键生成API文档**

FastAdmin中的一键生成API文档可以在命令行或后台一键生成我们API接口的接口测试文档，可以直接在线模拟接口请求，查看参数示例和返回示例。

//

**1、准备工作**

请确保你的API模块下的控制器代码没有语法错误，控制器类注释、方法名注释完整，注释规则请参考下方注释规则。请确保你的FastAdmin已经安装成功且能正常登录后台。请确保php所在的目录已经加入到系统环境变量，否则会提示找不到该命令。打开命令行控制台进入到FastAdmin根目录，也就是think文件所在的目录。

//

**2、常用命令**

// 一键生成API文档

php think api --force=true

// 指定<https://www.example.com>为API接口请求域名,默认为空

php think api -u <https://www.example.com> --force=true

// 输出自定义文件为myapi.html,默认为api.html

php think api -o myapi.html --force=true

// 修改API模板为mytemplate.html，默认为index.html

php think api -e mytemplate.html --force=true

// 修改标题为FastAdmin,作者为作者

php think api -t FastAdmin -a Karson --force=true

// 查看API接口命令行帮助

php think api -h

//

**3、参数介绍**

-u, --url[=URL] 默认API请求URL地址 [default: ""]

-m, --module[=MODULE] 模块名(admin/index/api) [default: "api"]

-o, --output[=OUTPUT] 输出文件 [default: "api.html"]

-e, --template[=TEMPLATE] 模板文件 [default: "index.html"]

-f, --force[=FORCE] 覆盖模式 [default: false]

-t, --title[=TITLE] 文档标题 [default: "FastAdmin"]

-a, --author[=AUTHOR] 文档作者 [default: "FastAdmin"]

-c, --class[=CLASS] 扩展类 (multiple values allowed)

-l, --language[=LANGUAGE] 语言 [default: "zh-cn"]

//

**4、注释规则**

在我们的控制器中通常分为两部分注释，一是控制器头部的注释，二是控制器方法的注释。

(1)、控制器注释(名称 描述 示例)

@ApiSector API分组名称 (测试分组)

@ApiRoute API接口URL，此@ApiRoute只是基础URL (/api/test)

@ApiInternal 忽略的控制器,表示此控制将不加入API文档 无

@ApiWeigh API方法的排序,值越大越靠前 (99)

(2)、控制器方法注释(名称 描述 示例)

@ApiTitle API接口的标题,为空时将自动匹配注释的文本信息 (测试标题)

@ApiSummary API接口描述 (测试描述)

@ApiRoute API接口地址,为空时将自动计算请求地址 (/api/test/index)

@ApiMethod API接口请求方法,默认为GET (POST)

@ApiSector API分组,默认按钮控制器或控制器的@ApiSector进行分组 (测试分组)

@ApiParams API请求参数,如果在@ApiRoute中有对应的{@参数名}，将进行替换 (name="id", type="integer", required=true, description="会员ID")

@ApiHeaders API请求传递的Headers信息 (name=token, type=string, required=true, description="请求的Token")

@ApiReturn API返回的结果示例 ({"code":1,"msg":"返回成功"})

@ApiReturnParams API返回的结果参数介绍 (name="code", type="integer", required=true, sample="0")

@ApiReturnHeaders API返回的Headers信息 (name="token", type="integer", required=true, sample="123456")

@ApiInternal 忽略的方法,表示此方法将不加入文档 无

@ApiWeigh API方法的排序,值越大越靠前 (99)

//

**5、标准范例**

<?php

namespace app\api\controller;

/\*\*

\* 测试API控制器

\*/

class Test extends \app\common\controller\Api

{

// 无需验证登录的方法

protected $noNeedLogin = ['test'];

// 无需要判断权限规则的方法

protected $noNeedRight = ['\*'];

/\*\*

\* 首页

\*

\* 可以通过@ApiInternal忽略请求的方法

\* @ApiInternal

\*/

public function index()

{

return 'index';

}

/\*\*

\* 私有方法

\* 私有的方法将不会出现在文档列表

\*/

private function privatetest()

{

return 'private';

}

/\*\*

\* 测试方法

\*

\* @ApiTitle (测试名称)

\* @ApiSummary (测试描述信息)

\* @ApiSector (测试分组)

\* @ApiMethod (POST)

\* @ApiRoute (/api/test/test/id/{id}/name/{name})

\* @ApiHeaders (name=token, type=string, required=true, description="请求的Token")

\* @ApiParams (name="id", type="integer", required=true, description="会员ID")

\* @ApiParams (name="name", type="string", required=true, description="用户名")

\* @ApiParams (name="data", type="object", sample="{'user\_id':'int','user\_name':'string','profile':{'email':'string','age':'integer'}}", description="扩展数据")

\* @ApiReturnParams (name="code", type="integer", required=true, sample="0")

\* @ApiReturnParams (name="msg", type="string", required=true, sample="返回成功")

\* @ApiReturnParams (name="data", type="object", sample="{'user\_id':'int','user\_name':'string','profile':{'email':'string','age':'integer'}}", description="扩展数据返回")

\* @ApiReturn ({

'code':'1',

'mesg':'返回成功'

\* })

\*/

public function test($id = '', $name = '')

{

$this->success("返回成功", $this->request->request());

}

}

//

**6、常见问题**

如果控制器的方法是private或protected的，则将不会生成相应的API文档。如果注释不生效，请检查注释文本是否正确。

--

**五、一键管理插件**

FastAdmin中的插件可以通过命令行快速的进行安装、卸载、禁用和启用。

//

**1、准备工作**

请确保你的FastAdmin已经能正常登录后台。请确保php所在的目录已经加入到系统环境变量，否则会提示找不到该命令。打开命令行控制台进入到FastAdmin根目录，也就是think文件所在的目录。

//

**2、常用命令**

// 创建一个myaddon本地插件，常用于开发自己的插件时使用

php think addon -a myaddon -c create

// 刷新插件缓存，如果禁用启用了插件，部分文件需要刷新才会生效

php think addon -a example -c refresh

// 远程安装example插件

php think addon -a example -c install

// 卸载本地的example插件

php think addon -a example -c uninstall

// 启用本地的example插件

php think addon -a example -c enable

// 禁用本地的example插件

php think addon -a example -c disable

// 升级本地的example插件

php think addon -a example -c upgrade

// 将本地的example插件打包成zip文件

php think addon -a example -c package

//

**3、常见问题**

付费插件请直接在后台插件管理或FastAdmin官方插件市场下载，然后进行离线安装。更多一键管理插件的参数请使用php think addon --help查看。

--

**六、一键安装FastAdmin**

FastAdmin可以在命令行使用命令快速的一键安装或重新安装。

//

**1、准备工作**

请确保你的数据库存储引擎支持innodb引擎，如果不支持将无法正常安装FastAdmin。请确保php所在的目录已经加入到系统环境变量，否则会提示找不到该命令。打开命令行控制台进入到FastAdmin根目录，也就是think文件所在的目录。

//

**2、常用命令**

// 一键安装FastAdmin

php think install

// 配置数据库连接地址为127.0.0.1

php think install -a 127.0.0.1

// 配置数据库用户名密码

php think install -u root -p 123456

// 配置数据库表名为dbname

php think install -d dbname

// 配置数据库表前缀为ff\_

php think install -r ff\_

// 强制重新安装FastAdmin

php think install -f 1

//

**3、常见问题**

更多一键安装FastAdmin的参数请使用php think install --help查看。

--