

ThinkPHP 6 快速开始

快速开始使用 ThinkPHP 6 进行项目开发

创建 ThinkPHP 项目

使用 docker 环境在本地创建 ThinkPHP 项目

- docker run -d
--name php_apache_server \
-v \$PWD/~/var/www/html \
-p 8002:80 \
php:8.0.3-apache
- docker start php_apache_server
- docker exec -it php_apache_server /bin/bash



创建 ThinkPHP 项目

装修 docker 的环境

- apt-get update
- apt-get install git vim zip unzip
- apt-get install libpng-dev
- docker-php-ext-install gd # gd 扩展
- apt-get install libzip-dev
- docker-php-ext-install zip # zip 扩展

创建 ThinkPHP 项目

真 · 创建 ThinkPHP 项目

- `curl -sS https://getcomposer.org/installer | php`
- `php composer.phar \`
 `create-project tophink/think lane`
- `php ../composer.phar install`
- `git init` # 初始化 GIT
- `sudo chown -R idea:idea lane`

服务器部署项目

及时测试代码是有效抑制BUG的技巧

- 云服务部署，域名解析，服务器备案
- 本地开发，本地部署

目录规划

清晰的规划会让生活充满希望

- controller
- middleware
- model
- service
- validate

增删改查

开始 CRUD 的枯燥生活

- 增加(Create)、读取查询(Retrieve)、更新(Update)和删除>Delete)
- 直接使用 DB 类操作数据
- 使用模型类操作数据

资源路由

RESTFul 是不错的选择

- 架构级别的决策
- HTTP 协议的最佳实践
- 清晰易懂的 API 风格

- 强制路由

验证客户端输入

谨慎处理细节是避免BUG的内功心法

- 场景验证

没错，会一个方法就行，选择多了，重构也困难。

不够用了在想办法。

- 验证异常处理

控制器中间件

小试一下面向切面编程

- 实现登录验证

到这里 你已经可以自己玩起来了