

Ant 扩展框架手册

1. 框架介绍

2. 文件结构

3. 快速入门

4. 类方法介绍

1. 框架介绍

Ant 框架是一个用 c 写的 php 扩展框架，目前支持：

1. 前端视图页面的操作：如分页，布局文件等.
2. 业务逻辑处理操作.
3. 提供了 php 原生扩展数据库操作.
4. 提供了 pathinfo 及默认 url 路由样式
5. 微服务
6. 微内核

后期支持功能：

1. 高性能的 rpc.
2. 支持数据库的并发查询.

3. 支持数据库原生协议的分库，分表操作.
4. 看情况考虑加入 **ORM**.(个人不建议)

2. 文件结构

Ant

```
| - apps-----应用目录
    | - module-----模块目录
        | - apis-----业务逻辑对外接口目录
        | - controllers-----控制器目录
        | - views-----视图目录
        | - models-----模型目录
        | - services-----服务目录
    | - logs-----日志目录
    | - publics-----静态资源目录
    | - vendors-----第三方引用库
    | - config.inc-----配置文件
    | - index.php-----入口文件
```

3. 快速入门

第一步：配置 config.inc 文件，修改 Ant\Config::initDb 信息.

第二步：在 **apps** 目录下创建模块，目录结构跟之前介绍的文件结构一样.

第三步：在 `module/controllers` 目录下创建控制器文件

`index_controller.inc`，内容如下：

```
<?php
    class IndexController extends Ant\AppController {
        public function indexAction() {
            $this->render();
        }
    }
```

第四步：在 `module/views` 目录下创建视图，视图文件在

`index --> index.inc`，完整路径：`module/views/index(控制器名字)/index.inc`，内容如下：

```
<?php echo "hello, ant"; ?>
```

第五步：nginx 或 apache 配置到框架根目录，然后访问请求：

<http://127.0.0.1/index.php?module=module&controller=index&action=index>，与普通的 php 框架无异。

到此，我们的快速入门就完成了，需要查看更详情的教程，请看写的 demo 程序(还未完成)。

4. 类方法介绍

4.1 Ant\Application

作用

入口应用类，主要用于请求信息，路由的初始化工作。

属性

`Protected _header`

保存设置的响应头部信息

`Protected _content`

保存视图内容

Protected `_module`

当前请求模块名称

Protected `_controller`

当前请求控制器名称

Protected `_action`

当前请求活动名称

Protected `_route_default_obj`

初始化对应的 `default` 路由对象

Protected `_route_pathinfo_obj`

初始化对应的 `pathinfo` 路由对象

方法

Public `__construct()`

Public `getAllSession(string)`

Public `getSession()`

Public `isGet()`

Public `isPost()`

Public `module()`

Public `controller()`

Public `action()`

Public `getRoute()`

Public `display()`

4.2 Ant\Config

作用

配置框架基本信息及自定义的配置信息.

属性

Protected _configs

保存框架配置信息

Protected _dbs

保存数据库配置信息

方法

Public initConfig(array)

Public getConfig(string)

Public setConfig(string, string)

Public initDb(array)

Public setDb(string)

Public getDb(string, string)

4.3 Ant\AppController

作用

作为控制器的父类，主要用于身份认证及权限通行.

属性

无

方法

Public __construct(object application, object view)

4.5 AntController

作用

控制器抽象类，作为 Ant\AppController 父类

属性

_view_obj

视图对象

_app_obj

Application 对象

方法

Public getUserInfo()

Public render(string file_name, string path, string
error_page)

Public message(string message, string tag)

Public query(string key)

Public form(string key)

Public isGet()

Public isPost()

Public module()

Pulbic controller()

Public action()

Public setResponseHeader()

Public redirect(string url, bool full)

Public assign(string name, zval* value)

4.4 Ant\Core

作用

微内核，提供文件加载方法.

属性

无

方法

Public static import(string fileName, [string path])

4.5 AntRoute

作用

路由接口

属性

无

方法

Public url(string url, bool full)

Public parse()

4.6 AntARoute

作用

路由抽象类，实现 **AntIRoute** 接口.

属性

`_application_obj`

Application 对象

`_url_params`

Url 参数

方法

无

4.7 AntRouteDefault

作用

默认路由类，继承 **AntARoute** 类.

属性

无

方法

`Public __construct(object applicationObj)`

4.8 AntRoutePathinfo

作用

Pathinfo 路由类, 继承 **AntARoute** 类.

属性

无

方法

Public __construct(object applicationObj)

Public generate(string url, bool full)

4.9 AntView

作用

视图类

属性

无

方法

Public render(string path)

Public viewVars(string key)

Public inflateFile(string file_name, string
module_path)

Public showMessage()

5.0 AntDatabase

作用

数据库类

属性

`_conn_id`

连接句柄

`_query_id`

查询句柄

方法

`Public __construct(string host, string user, string password, string dbname, string charset)`

`Public query(string sql)`

`Public insert()`

`Public update()`

`Public getRow()`

`Public getRows(string type)`

`Public getField(string type)`

`Public getLastId()`

`Public begin_transaction()`

`Public rollback_transaction()`

`Public commit_transaction()`