X PHPFramework Document

一.安装配置

I. php-fpm 配置方法:

使用 php-admin-values 命令，添加 auto_prepend_file 配置,加载框架__init__.php 文件

例如：在 php-fpm.conf 文件中添加

php_admin_value[auto_prepend_file]=*[FRAMEWORK_DIR]*/__init__.php

II. PHP 以 apache 模块方式运行

在 Vhost 块中添加,或者添加到.htaccess 中

php_admin_value auto_prepend_file *[FRAMEWORK_DIR]*/__init__.php

二.相应网站目录

/ 网站程序所在目录
/data 网站数据目录
/data/cache 缓存数据目录
/theme 网站主题目录
/www 网站 webroot 目录
/lib 用户自定义库,一个文件只包*[一个文件名同名]*的类
/dbm 数据库查询库，一个文件只包含一个 *[dbm 和文件名组成类名]*的类

三.二级网站定义方法

在/www 目录下定义一个文件夹，同时在/theme 目录下定义同名文件夹，然后二级网站二级域名 webroot 指向该文件夹
类似/www/*[二级网站目录]*/*[组件名]*/index.php

四.组件创建方法

在 webroot 指向的目录新建一个目录（目录名建议为该组件名），然后创建一个 index.php 文件，该文件应当有一个 index 类
index 类继承 x 类（如果不需要使用框架方法或函数可以不继承）
该类必须定义一个名为 defaults 的 public 方法,该方法作为该类默认访问方法
可选特殊 public 方法：declare_instance_list, 该方法类似构造函数
可选特殊 static public 属性：$function_list，该属性为一个数组，定义了当前组件可使用---外部方法(见下)---

五.外部方法

在组件下创建一个与外部方法同名的文件，该文件包含一个外部方法名的函数，函数一个参数为,该参数将作为组件类的方法指针(类似$this 指针)
外部方法可以通过函数参数方法本组件下的所有外部方法与 index 类的方法和属性
外部方法自定义参数从第二个开始

六.模板使用方法

有以下内置属性
在组件 index 类中
$this->T->name 为模板名称，对于/theme/[二级网站目录]/相应文件名,模板后缀为.htm
$this->T->noecho bool 值，设置是否有输出，false 为默认有输出
$this->T->notpl bool 值，设置是否有模板,如果该值与$this->T->name 都为设定，将为 true
$this->T->no_cache_html bool 值 设置是否生成 html 静态文件，true 为默认不设置
$this->T->record_num 总记录数，分页相关
$this->T->current 当前页，分页相关
$this->T->page_num 总页数，分页相关
$this->D 该属性保存了模板中变量的对象，模板中变量为 {$a}的设置方法为 $this->D->a = 'this value'
在组件外部方法中使用方法的第一个参数代替这里的指针$this
模板解析标志符号 { 和 }
模板中显示变量方法 {$var}
模板中设置变量方法 {set var=value}
模板中调用函数 {func functionname($arg1,$arg2.....)} 和 PHP 函数调用方法一样
模板中内置函数有：show_page_num() 显示内置分页
模板中 URL 设置方法 {url=DIR|page|param} dir 为组件所在目录，page 为组件内文件名,param 为参数,支持变量，最后两个参数为可

选

七.重定向页面

在组件方法或用户库类中使用 $this->class_redirect($part_name,$page,$params)进行重定向

八.用户类，/lib 文件夹下的类，建议继承 librdm

librdm 类实现了 session 与 cookie 的面向对象访问

九.数据库操作类 /dbm 文件夹下的类，必须继承 libdbm

1.数据库操作实例

libdbm:db 访问方法 $this->db

2.数据库表操作实例

libdbm::db::tabname 访问方法 $this->db->tabname

3.数据库操作实例拥有方法

a. dba::query($sql) 查询，返回资源符
b. dba::assoc() 获取查询资源中的一条结果,字段名为 KEY
c. dba::row() 获取查询资源中的一条结果,数字索引
d. dba::count_rows() 获取查询资源中总条数
e. dba::fetch($sql) 查询并返回所有数据集，字段名为 KEY
f. dba::get_one_row($sql) 查询并返回一行数据,字段名为 KEY
g. dba::get_one($sql) 返回第一行第一列数据
h.dba::insert_id() 返回上一次 INSERT ID
i. dba::affected_rows() 取得前一次 MySQL 操作所影响的记录行数
j.dba::free() 释放上一次查询资源
k.dba::close() 关闭 mysql 连接

4.数据库表操作实例拥有方法

a.dba_func::columnus() 获取当前数据库表对象的表字段列表
b.dba_func::columnus_list_sql() 将当前表字段名转换成 SQL 可用字符串
c.dba_func::auto_select($limit, $start) 自动获取当前表数据集
d.dba_func::primary() 返回当前数据库表主键名
e.dba_func::primary_select_by_in($in_arr) 查询主键等于指定数组中的所有值的集合
f. dba_func::primary_select($value) 查询主键对于指定值的一行数据
g. dba_func::primary_delete($value) 删除主键等于指定值的一行
h.dba_func::primary_update($value, $array) 更新主键等于数组 KEY 为字段名的所有值
i. dba_func::auto_select_count($where) 获取以$where 语句为条件的数据集合总数
j.dba_func:: get_one_field($field,$limit,$start) 获取字段为$field 的列的所有数据

十.Ajax 上传的 JSON 数据处理

1.统一输出/接收 JSON 数据格式

{status:'',message:'',data:''}

status：表示状态，目前只有两种状态, 1 正常, 0 错误
message： 返回提示信息
data：需要返回的数据

2.设置返回 JSON 数据,只需要调用

$this->set_ajax($status, $message,$data)， 外部函数用函数第一个参数代替$this

3.获取 JSON 数据

$this->get_ajax_data()
将返回一个包含数据的对象

4.检查接收的 JSON 数据中是否存在某一值

$this->check_ajax_data($idx)

十一. PHP 接收数据

1.$_REQUEST 值访问,

实例： X::R

例如：$_REQUEST['name']在对象中的访问方法:

$this->R->name->v 原值

$this->R->name->html 将 ' " < > 转换成 HTML 元字符

$this->R->name->trim 去掉首尾空白字符

$this->R->name->removal 去掉单引号和双引号

$this->R->name->sal 转义单引号

2.$_GET

实例: X::R::G

例如：$_GET['name'] 在对象中的访问方法

$this->R->G->name->v 原值

$this->R->G->name->html 将 ' " < > 转换成 HTML 元字符

$this->R->G->name->trim 去掉首尾空白字符

$this->R->G->name->removal 去掉单引号和双引号

$this->R->G->name->sal 转义单引号

3.$_POST

实例 X::R::P

类似于$_GET 访问方法使用

$_POST 使用 $this->R->P->name

4. cookie 控制

实例 X::cookie::name

I.访问方法: $this->cookie->name, 将返回$_COOKIE['name']的值

II.设置,实例 X::cookie::name()

$this->cookie->name() 获取名为 name COOKIE 设置对象

$this->cookie->name()->value 设置 cookie 值

$this->cookie->name()->expire 设置 cookie 有效期

$this->cookie->name()->path 设置 cookie 作用路径

$this->cookie->name()->domain 设置 cookie 作用域

$this->cookie->name()->secure 设置是仅在 https 传送

$this->cookie->name()->httponly 设置 cookie 只作用与 http

$this->cookie->name()->set() 设置 cookie 值

5. session 控制

$this->session->name 设置或获取 名为 name 的 session