Smarty 模板引擎

作者：西岭
Email：xiling3105@163.com
指导：燕十八
出自：布尔教育--高级IT工程师培训

学习方法

HTML, PHP基础巩固,面向对象,MySQL
以上为理论课程,很难从0摸索出来.
后面的课程多为工具性质的产品.

1.1 学习心态:

- 不要怕，你肯定可以掌握
- 要敢于摸索,不要被动等老师讲
- 最快的办法就是试用

1.2 具体步骤:

1. 下载解压
2. 看官方手册/demo/INSTALL/README ,快速搭建一个例子
3. 手册增加例子的复杂度

1.3 错误方式:

先baidu,google一堆网页,或者下载视频.
因为
+ 网页上的东西,绝大部分是抄的手册,而且相互转载
+ 网页相对于手册是落后的.
+ 资料找的越多,越学不进去.

前后端

PHP和HTML都由一个人来写时,可能是这样的

```
<?php
    $sql = 'select ....'; //查询等等
    $title = '今天天气不错';
?>
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Document</title>
    </head>
    <body>
        <h1><?php echo $title;?></h1>
    </body>
</html>
```

PHP和HTML都由两个人来写时,可能是这样的
xx.php

```
$title = '标题';
$content = '这是内容';
//引入模板文件
include(./1.html);
```

xx.html

```
<h1><?php echo $title; ?></h1>
<p><?php echo $content; ?></p>
```

但是,但是,但是, 前端人员,不要看到PHP标签,怎么办?
我们的代码变成这样子:

```
$title = '标题';
$content = '这是内容';
//引入模板文件
include(./1.html);
```

前段程序猿

```
<h1>{$title}</h1>
<p>{$content}</p>
```

输出效果，很显然不是我们要的样子

矛盾所在:

前端想要的: `{$title}`

PHP 需要的: `<?php echo $title; ?>`

解决办法:

用程序把前端想要的标签,转换为PHP需要的标签,这样2者都可满足

程序思路 :

把 `{$` 替换为 `<?php echo $`

把 `}` 替换为 `;?>`

file_get_contents、file_put_contents、str_replace

```
function comp($temp) {
    $html = file_get_contents($temp);
    $html = str_replace('{$', '<?php echo $', $html);
    $html = str_replace('}', ';?>', $html);
    $compfile = $temp.'.php';
    file_put_contents($compfile, $html);
    return $compfile;
}

$sql = 'select ....'; //查询等等
$title = '今天天气不错';
include( comp('./xx.html') );
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
    <title>Document</title>
</head>
<body>
    <h1>{$title}</h1>
</body>
</html>
```

mini模板类

把刚才编译模板的函数封装成模板类

```
class Mini {
    public function comp($temp) {
        $html = file_get_contents($temp);
        $html = str_replace('{$', "<?php echo \$this->data['", $html);
        $html = str_replace('}', "'];?>", $html);
        $compfile = $temp.'.php';
        file_put_contents($compfile, $html);
        return $compfile;
    }
    public function display($temp) {
        include( $this->comp($temp) );
    }
}

$sql = 'select ....'; //查询等等
$title = '今天天气不错';
$mini = new Mini();
$mini->display('./xx.html');
```

错误如下:

Notice: Undefined v ariable: title in xx.html.php on line8

错误原因:

在于PHP变量的作用域!

.php文件在display() 方法里被包含

当然读不到全局变量$title

解决办法:

把外界变量传入对象的属性中,让被包含的.php文件,引用对象的属性值

```
class Mini {
    protected $data = array();
    // 把外界变量装进data数组
    public function assign($k,$v) {
        $this->data[$k] = $v;
    }
    public function comp($temp) {
        $html = file_get_contents($temp);
            // 注意 下行引用自身的data属性
        $html = str_replace('{$', "<?php echo \$this->data['", $html);
        $html = str_replace('}', "'];?>", $html);
        $compfile = $temp.'.php';
        file_put_contents($compfile, $html);
```

```
            return $compfile;
        }
        public function display($temp) {
            $compfile = $this->comp($temp);
            include($compfile);
       }
    }

    $mini = new Mini();
    $mini->assign('title' , $title);
    $mini->display('./xx.html');
```

编写类，模拟smarty的实现原理
总结: Mini模板类,其实就是一个Mini Smarty类.
可以推荐出Smarty模板类的用法:
* 引入Smarty类
* 实例化Smarty类
* assign赋值
* display输出

学习sarty

基本使用

复制核心文件到项目内
Smarty 的引入
Smarty 是一个类, 引入过程与普通的类没有区别
1. include,require包含此文件
2. 实例化
基础配置（缓存目录，边界符.....）

```
require './libs/Smarty.class.php';
$smarty = new Smarty;
$smarty->template_dir = './dir';//设置模板路径
$smarty->left_delimiter = '<{'; //设置左边界符
$smarty->right_delimiter = '}>';//设置右边界符
$smarty->caching = true;//是否开启缓存
$smarty->cache_lifetime = 120; //缓存的生命周期(秒)
$t = '北京雾霾';
$smarty->assign('t',$t);
$smarty->display('2.html');
```

基础用法

Smarty 的赋值

1: `$smarty->assign('key',value);`
发生的变化 `$smarty->_tpl_vars[key] = value;`
和我们自己写的类的处理方式类似
2: 如果 `assign($arr)` 中的第一个参数是数组
则会循环数组,把数组的每一个单元追加到_tpl_vars属性上

模板中如何引用变量

- 字符串型,数值型, 直接通过 `{$标签名}` 来引用 例: {$title}
- 数组变量,用 `$标签名.键或$标签名[键]` 来引用数组单元
  例: `{$stu.name} {$stu[age]}`

  > *建议用后者,即中括号语法,兼容性更强*

- 对象
  用 `$标签名->属性名` 来引用对象的属性值
  用 `$标签名->方法()` 来调用对象的方法的返回值
- smarty中的系统变量
  有些系统变量,不需要assign(), 即可使用, 引用时以 `$smarty` 开头
  例:

```
$smarty.now , 被 解析成 time();
$smarty.get.key ---> $_GET[key]
$smarty.const.常量名 ---> echo 常量名
```

- 从配置文件得到的变量 配置文件可以用来存储常用且很少变的数据,比如网站名,备案号 通过配置文件得到这些信息,不必去读数据库,可以省一些数据库的开销. 配置文件的写法

例:xxx.conf
配置项1=值1
配置项2=值2
配置文件的载入

`{Config_load file="xxx.conf"}`

引用配置选项的值

`{$smarty.config.配置项} 或者 {#配置项#}`

总结:模板中的变量有3种来源

1: assign赋值得到的变量 , 存储在_tpl_vars属性中

2: $smarty系统变量, 对于cookie,session,get,post,$_SERVER等信息,存储在_smarty属性中

Smarty会自动捕捉,并保存起来,形成系统变量,可以直接用标签来引用.

3: 从配置文件读取的变量 , 存储在_config属性中;

定界符冲突的问题

如果smarty用定界符,比如 {,},

此定界在js,css里都有很可能碰到,如果碰到,会当成smarty标签来解析,进而引发错误发生.

1:换定界符,如 {> <}

2:用literal标签,"原义","字符意义的","无夸张的"

例: {literal}h1{background:gray}{/literal}

判断、循环、运算

* 提问: *模板的职责,在于输出数据.

基于此,模板中不应该有if/else,等逻辑相关的东西.

为什么 还要在模板中加逻辑控制的功能?

场景:

某商城,针对vip会员 和 普通会员, 显示不同的效果

如:

VIP: 1.87元

普通: 399.87元

如果模板上没有任何逻辑控制,那么只能在PHP上做逻辑判断.

只能如下方式来完成:

```
// xx.php
if($lev == 'vip') {
$smarty->display('vip.html');
} else {
$smarty->display('comm.html');
}
```

- 判断

```
{if $ss === xxx}
    xxxx
{else if $xx ===xx}
    xxxxx
```

```
{else}
    xxxx
{/if}
```

- 循环

```
//xx.php
$news = array(
    array('id'=>1,'title'=>'释永信离婚','pubtime'=>1345213687),
    array('id'=>2,'title'=>'丁俊晖又输了','pubtime'=>1425632587),
    array('id'=>3,'title'=>'宁泽涛有2亿老婆','pubtime'=>125467582),
    array('id'=>4,'title'=>'万达快被清算了','pubtime'=>123654687),
);
$smarty->assign('news' , $news);
```

- 模板

```
{foreach $new as $k=>$v}
    {$v.title}
    {$v.id}
{/foreach}
```

- 模板中的运算符

```
{$xxx + 2}
{$xxx * 2}
{if $nu > 3}
```

不推荐在模板中这样做,这不是模板的职责

变量调节器及模板编译的特点

可以把标签变量的值,做一些修改.
例: `{$intro|upper}` ,会把 `$info` 的内容转换为大写
原理 : 把 `$intro` 作为参数,传给upper调节器对应的函数,
并显示该函数值,而不是 `$intro` ;
`$intro` 标签变量本身会当成调节器的第1个参数自动传入,
如果需要传更多参数,在调节器后面,用' :'隔开更多参数.
例 `{$new s| truncate:7:"..."}`

- 变量调节器
date_format [格式化日期]
default [默认值]
escape [编码]
indent [缩进]
lower [小写]
nl2br [换行符替换成 <\br />]
replace [替换]
strip [去除(多余空格)]
strip_tags [去除html标签]
upper [大写]
示例:
{$smarty.now|date_format:'%Y-%m-%d %H:%M:%S'}

- 模板编译的特点:

Smarty的一个特点: 先编译成.php文件,再执行该 PHP
Smarty在第一次运行的时候,稍慢,包含了编译模板+包含执行
后面的运行中速度会快,因为直接包含PHP文件.

问: smarty什么时间重新编译模板?
答: 根据模板修改的时时间.

创建模板-->编译模板-->修改模板-->再次编译

如果想强制编译: 比如在开发过程中
可以force_compile=true 强制编译

display和fetch的区别

display 调用的ftech方法
即: display() = echo fetch();
推测:

```
$smarty->display('xx.html'); // 输出
$smarty->fetch('xx.html'); // 不输出
echo $smarty->fetch('xx.html'); // 输出
```

直接看源码，找到元凶:

```
// file Smarty.class.php 第831行左右
public function display($template = null, $cache_id = null, $compile_id
```

```
 = null, $parent = null) {
// display template
    $this->fetch($template, $cache_id, $compile_id, $parent, true);
}

// 806行左右
public function fetch() {
// ....
return $_template->render(true, false, $display);
}
```

```
// file sysplugins/smarty_internal_template.php 302行左右
// display or fetch
if ($display) {
// ...
echo $content;
return '';
} else {
// ...
return $content;
}
```

包含子模板

头部尾部的文件引用等，很实用;
{include file='./xxx.html'}

缓存

1:首选打开缓存配置项 `$smarty->caching = true;`
2:缓存生命周期的配置选项: `$smarty->cache_lifetime = 整数秒`
3: `$smarty->isCached()` 判断是否有缓存,如果有缓存,则用缓存，避免数据交互操作

```
if( !$smarty->isCached('xx.html') ) {
    $rs = mysql_query('select * from art limit 3' , $conn);
    $art = mysql_fetch_assoc($rs);
    $smarty->assign('art' , $art);
    echo 'from mysql';
}
$smarty->display('07.html');
```

单模板多缓存

1个模板还可以根据其他参数,缓存多个结果,比如商品页面根据商品id来为每个商品缓存一个页面.
原则: 凡是能够影响到页面内容的参数,就要根据此参数影响cached_id
这样,不同参数才能对应不同的缓存内容

用法:
Display(模板名[,缓存名]);
同理, 在判断某个模板是否已被缓存的时候,也需要注意,
要传一个缓存名.
即 isCached(模板名,缓存名) 才能合理判断.

```
$art_id = $_GET['art_id'];
if( !$smarty->isCached('07.html' , $art_id) ) {
    $rs = mysql_query('select * from art where art_id='.$art_id , $conn
);
    $art = mysql_fetch_assoc($rs);
    $smarty->assign('art' , $art);
    echo 'from mysql';
}
$smarty->display('07.html',$art_id);
```

缓存判断原理
当前时间 - 缓存生成时间 > 生命周期

局部缓存

不缓存标签
可以以不缓存的方式assign()变量。
适用于单个标签不缓存。
例:

```
php
- $smarty->assign('foo',$foo,true); //第三个参数 true 不缓存
```

nocache 标签 标签
在模板中,不需要缓存的地方,用`{nocached}{/nocached}`包住.
适用于大段的不缓存效果
html模板
例:`{nocache}{$smarty.now}{/nocache}`

模板调用PHP函数

- 模板中用{insert name=xx} 直接调用PHP中的insert_xx的函数
//不缓存

模板引擎之殇

某些编程语言，没有模板不行，但是，PHP可以不用

```
print("<html>")
print("<h1>")
print("我是标题")
print("</h1>")
print("</html>")
```

这种情况下,python与html混杂在一起.
没有模板支持,前端制作人员不懂python很难工作.
所以需要让前端用模板的形式来写页面,

而python再用专门的一个类去处理模板,并输出.
常用的python模板类有
jinja2，cheetah，mako，webpy，bottle，tornado
思考: 相比python,为什么php可以不用模板?

php本身就是一种标签语言

php代码可以嵌套在html中
混杂在H TML代码中smarty标签和PH P代码.
对于web前端开发人员来说,
本质上

```
<p>{$t}</p>
和
<p><?php echo $t;?></p>
```

对于前段人员，没有本质的区别,
但是，性能却降低了;
又但是，缓存还是有价值的
模板的副作用:
解析编译，消耗性能
增多了很多变量 页面内的变量,都要赋值到smarty对象-->_tpl_vars 属性上, 多了一个变

量的副本
因此:
ci,yii2 抛弃引擎，直接嵌套

还有什么爱你的理由呢?
* 缓存(但有很多更高效的缓存工具,如memcached)
* 面试会问到,有些古老的项目还会用到.
* 一些框架(TP,laravel),也有自己的模板引擎. 和Smarty的模板语法触类旁通,降低学习成本.

MVC和smarty的关系

model 数据处理层
controller 业务逻辑层
view 视图展示层

```
function add($a,$b){
    return $a+$b;
}


$a = $_GET['a'];
$b = $_GET['b'];
if(!empty($a) && !empty($b)){
    $res = add($a,$b);
}else{
    $res = '参数缺失，无法运算';
}

echo '<html><h1>'.$res.'</h1></html>';
```

MVC和smarty到底是个什么关系?