Tài liệu tham khảo

https://www.smarty.net/docs/en/what.is.smarty.tpl

https://www.smarty.net/docs/en/

https://groups.google.com/forum/#!topic/phpvietnam/VHpDo6Bc8dc

https://www.smarty.net/docs/en/installing.smarty.basic.tpl

**Requirements**

Smarty requires a web server running PHP 5.2 or greater.

**Ưu điểm:**

- $smarty->compile\_check = true // kiểm tra xem template đã có thay đổi gì không, nếu có, nạp dữ liệu mới, nếu false thì sẽ ko kiểm tra mà dùng file đã biên soạn, dù có xóa nội dung file cũng ko hề gì

$smarty->force\_compile = TRUE; // thiết lập chế độ k dùng file biên soạn sẵn

This setting overrides [*$compile\_check*](https://www.smarty.net/docsv2/en/variable.compile.check.tpl). By default this is FALSE. This is handy for development and [debugging](https://www.smarty.net/docsv2/en/chapter.debugging.console.tpl). It should never be used in a production environment. If [*$caching*](https://www.smarty.net/docsv2/en/variable.caching.tpl) is enabled, the cache file(s) will be regenerated every time.

$smarty->setCompileDir('libs/smarty-3.1.30/templates\_c/');

- $smarty->setTemplateDir('./pages'); // nếu enable security thì các file tpl phải được khai báo trong folder pages.

**Security**

- $smarty->enableSecurity(); // enable default security

+ $php\_handling :

Smarty::PHP\_PASSTHRU -> echo PHP tags as they are ( default )

Smarty::PHP\_QUOTE -> escape tags as entities

Smarty::PHP\_REMOVE -> remove php tags

Smarty::PHP\_ALLOW -> execute php tags -> dùng new SmartyBC và dùng {php}{/php}

+ $trusted\_dir : là array chứa những thư mục có thể dùng include\_php trỏ đến file trong thư mục đó. DÙng với SmartyBC

+ $php\_functions = array('isset', 'empty', 'count', 'sizeof', 'in\_array', 'is\_array', 'time',) // Những function được dùng trong tpl, NULL là disabled all, empty là allowed all

+ $php\_modifiers = array('escape', 'count', 'nl2br',); // Những hàm có thể dùng trong tpl như {$name|escape}, NULL là disabled all, empty là allowed all

+ *$trusted\_uri* is an array of regular expressions matching URIs that are considered trusted.

The expression '#https?://.\*smarty.net$#i' would allow accessing the follwing URIs:

http://smarty.net/foo

http://smarty.net/foo

http://www.smarty.net/foo

http://smarty.net/foo

https://foo.bar.www.smarty.net/foo/bla?blubb=1

but deny access to these URIs:

http://smarty.com/foo (not matching top-level domain "com")

ftp://www.smarty.net/foo (not matching protocol "ftp")

http://www.smarty.net.otherdomain.com/foo (not matching end of domain "smarty.net")

**Template Inheritance**

**Function**

<ul>

{for $foo=1 to $loop|count}

<li>{$foo}</li>

{forelse}

no iteration

{/for}

<ul>

{foreach $loop2 as $value}

<li>{$value.id}

{$value.name}

</li>

{foreachelse}

.. no results ..

{/foreach}

</ul>

<br>

{function menu level=0}

<div>{$abc}</div>

<ul class="level{$level}">

{foreach $data2 as $entry}

{if is\_array($entry)}

<li>{$entry@key}</li>

{menu data2=$entry level=$level+1}

{else}

<li>{$entry}</li>

{/if}

{/foreach}

</ul>

{/function}

{$menu = [

'item1',

'item2',

'item3' => [

'item3-1',

'item3-2',

'item3-3' => [

'item3-3-1',

'item3-3-2'

]

],

'item4'

]

}

{menu data2=$menu abc=123}

<div>sdsadasd {ldelim}sadasd{rdelim}</div>

Today's date is

{nocache}

{$smarty.now|date\_format}

{/nocache}

class Bar {

$property = "hello world";

}

$smarty = new Smarty();

$smarty->registerClass("Foo", "Bar");

{\* Smarty will access this class as long as it's not prohibited by security \*}

{Bar::$property}

{\* Foo translates to the real class Bar \*}

{Foo::$property}

// put this in your application

function protect\_email($tpl\_output, Smarty\_Internal\_Template $template)

{

$tpl\_output =

preg\_replace('!(\S+)@([a-zA-Z0-9\.\-]+\.([a-zA-Z]{2,3}|[0-9]{1,3}))!',

'$1%40$2', $tpl\_output);

return $tpl\_output;

}

// register the outputfilter

$smarty->registerFilter("output","protect\_email");

$smarty->display("index.tpl');

// now any occurrence of an email address in the template output will have

// a simple protection against spambots

**Nhược điểm:**

Thông báo lỗi khó debug, vì thông báo lỗi ko hiển thị số dòng ở file template mà hiển thị ở file đã biên soạn

Ngay sau khi biên dịch xong, Smarty vẫn cần có lời gọi hệ thống stat để kiểm  
tra sự thay đổi của template gốc.  Stat là lời gọi rất tốn kém tài  
nguyên. Và như thế nó đã làm giảm performance của compiled template so  
với pure PHP template. Khi truy xuất I/O, PHP vẫn phải gọi stat call  
nhưng riêng Smarty, nó cần thêm các lời gọi bổ sung để đảm bảo cho mô  
hình của nó hoạt động.

Chậm hơn so với pure PHP

Để sử dụng thì phải học các cú pháp của Smarty