Smarty Templates

Tiki uses the Smarty Template Engine to control themes (mostly the layout, and some design / layout logic).

"One of Smarty's primary design goals is to facilitate the separation of application code from presentation. Typically, the application code contains the business logic of your application, written and maintained in PHP code. This code is maintained by programmers. The presentation is the way your content is presented to the end user, which is written and maintained in template files. The Templates are maintained by template designers."

http://www.smarty.net/whyuse.php

Smarty is a "Template/Presentation Framework." It provides the designer with the opportunity to change the presentation of a website by defining variables and using logic (If/else) statements. It can be used for example to create WYSIWYCA ("what you see is what you can access") websites which show or hide things depending on permissions variables.

History

- The very first version of Tiki (October 2002) was using Smarty v2.2.
- Smarty version 3 was introduced in Tiki8 (2011)
- Smarty version 4 was introduced in Tiki25 (2022)
- Smarty version 5 was introduced in Tiki27 (2024)
    - https://github.com/smarty-php/smarty/discussions/920

Default Templates and Custom Templates

The default Smarty template files are in the folder /Templates as .tpl files. They can be edited with any text editor, but it is best to not edit the default version of these templates.

Since Tiki 15, if you want to modify a template, copy it and put the copy in the custom theme directory, e.g. themes /**custom_theme_name**/templates (it was /templates/styles/**custom_theme_name**/ prior to Tiki 15).

See Themes ("How To Create a Custom Theme") for more details.

Escaping variables used in Smarty

When modifying templates, it is important to check that variables that display output on screen be escaped except in certain circumstances, to act as a safeguard agains unfiltered user input being displayed back on the page. The question is when to escape and when not to escape? See http://www.smarty.net/docs/en/language.modifier.escape.tpl

- If the output of your template will be used in another template **as a variable that will be escaped there**, then there is no need to escape it the first time as it will lead to double escaping.
- If the variable is expected to possibly contain HTML, then it cannot be escaped otherwise the HTML will be displayed as text on screen. In those cases, it is the responsibility of the code that generated the HTML (e.g. the wiki parser to ensure that the HTML output is filtered), and the code of the input user interface to perform filtering in case any user enters any HTML - only trusted users should be allowed to enter any HTML).
- If the variable is used for redisplaying what the user entered **in an input text field**, then it cannot be escaped otherwise it will lead to double escaping the next time the user submits the form.
- In all other cases, the variable should be escaped using the escape variable.
    - Note that if the variable is to be used in a URL, then it must be escaped using the "url" method rather than just the standard method. See http://www.smarty.net/docs/en/language.modifier.escape.tpl for more information.

Tips

- use the **{literal} {/literal}** tag to escape Smarty parsing for a block of code, e.g. for a javascript
- **{* comment *}** is used for commenting in smarty
- **{tr}some text{/tr}** is used for strings in the UI that are intended to be translated into other languages (see Internationalization).
- use the **{wikiplugin} {/wikiplugin}** tag to include a wiki plugin
  e.g. for nesting trackerlist plugins, use in a trackerlist's template:
  {wikiplugin _name=trackerlist ignoreRequestItemId=y trackerId=X filterfield=Y filtervalue="{$your_var}"}{/wikiplugin}

  Note: when a plugin takes values in its **body**, e.g. PluginMouseover, this content should go between the opening and closing **wikiplugin** tags (this comes from the fact that smarty "block" objects have a body between the open and close tags):
  {wikiplugin _name=mouseover label='Hello' offsetx='4' offsety='15'}{$your_content}{/wikiplugin}
- add **&show_smarty_debug=1** to the real url to inspect smarty variables
  e.g. http://www.example.com/tiki-index.php?page=examplepagename&show_smarty_debug=1
- use **{wiki}wiki syntax here{/wiki}** to put wiki syntax in smarty templates

Samples

Wikiplugin group

The wiki plugin group can be used the same way if you need to set it without an alternative to the condition ({else})

```
{wikiplugin _name='group' groups='Admins'} I'm a member of the admins group {/wikiplugin}
```

However using {ELSE} like you would in the wiki syntax will break the code.
In that case of an alternative to a condition you need to write the condition as follow:

```
{if 'Admins'|in_group} I'm a member of the admins group {else} I'm not a member of the admins group {/if}
```

Using smarty modifier (number format) with data from a pluginList

You can't use directly the modifier
`|number_format`

in a smarty template on a formatted field from a pluginList. IE:

```
{$row.prix|number_format}
```

You need to use it on the direct value from the tracker :

```
{$row.tracker_field_catalogueDesBiensPRIX|number_format}
```

Use an URL parameter assigned to a smarty variable

With {$smarty.get.urlParameter} is possible to use an URL parameter directly into your smarty template.
IE: Using a list plugin you generate your URL with an URL parameter like:

```
<a href="holidays-camps-childrenList?holiday={$row.tracker_field_holidaysReference}">
```

that will generate
https://yourTiki.com/holidays-camps-childrenList?holidayReference=Christmas

In your smarty template you can reuse then the value to display it or embed a second wikiplugin list

```
{filter field="tracker_field_holidaycampsNameReference"
exact="{/literal}{$smarty.get.holiday}{literal}"}
```

Using the Tiki language preferences

You can use the selected language to control what is displayed or used in embedded wikiplugin (like an embedded wikiplugin List)

Using the language preferences

```
{if $prefs.language eq fr}Montre ca {else} Show this {/if}
```

See also

- PluginSmarty

More information

- http://www.smarty.net/crashcourse.php: Crash Course in Smarty
- http://www.smarty.net/manual/en/: Smarty Manual
- https://themes.tiki.org - a resource for Tiki theme designers.

Alias

- tpl
- Smarty
- Smarty Template