

µWeb - Feature #61

Template parser - default values

2011-05-02 11:54 - Jacko Hoogeveen

<b>Status:</b>	Closed	<b>Start date:</b>	2011-05-02
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Jan Klopper	<b>% Done:</b>	100%
<b>Category:</b>	TemplateParser	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Spent time:</b>	4.00 hours
<b>Description</b> The ability to assign a default value to a template. When the template parser finds a tag unused replace it with the default value if available.  Default values could be described in multiple formats  <b>inline:</b>  [<tag> default=<default value tag>]  <b>example:</b> [search_result default=<p>No search results found.</p>] [icon default=<img src='/icon_default.png'>]  <b>Django style (multiline):</b>  [<tag>] <default value tag> [END]  <b>example:</b>  [search_result] <p>No search results found.</p> [END]  [icon] <img src='/icon_default.png'> [END]			
<b>Related issues:</b> Related to µWeb - Feature #252: Template parser - if statements & for loops			
		Closed	2011-06-29

Associated revisions

Revision 3300:e85529599039 - 2012-01-10 18:03 - Elmer de Looff

Added {{ if }} statements to the templateparser. This allows richer expressions in templates. if, elif and else are properly supported. This completely resolves #252 and makes obsolete issue #61.

Revision 157:83666f9b5d14 - 2012-01-10 18:03 - Elmer de Looff

Added {{ if }} statements to the templateparser. This allows richer expressions in templates. if, elif and else are properly supported. This completely resolves #252 and makes obsolete issue #61.

History

#1 - 2011-05-26 21:05 - Jan Klopper

Maybe we can overload the function syntax.

[var|function|default]

If the var doesn't exist, see if the last function also does not exist, and if so use that string as the value.

It's a hack though, but I like the | as an OR.

Maybe || would be better though.

## #2 - 2011-05-27 00:13 - Elmer de Looff

- Status changed from New to Feedback

- Assignee set to Jan Klopper

| is already used as a pipe, pushing the result (of the previous) towards the (next) function.

We should not assign a second meaning to the pipe character. Double pipes are going to reduce readability significantly -- "is this an error, or a default value?"

The Django multiline default is an option, but then we're special casing the word END, which seems a tad hacky as well.

What I could see happen is a {default} syntax that follows the last function but precedes the closing bracket:

```
<p>This is a demonstration of the template default. Kittens are:</p>
<ul>
  <li>Cute</li>
  <li>[behaviour|lolcat{freaking annoying}]</li>
</ul>
```

If behaviour is specified, this would process behaviour with the lolcat template function and place it. If behaviour is left unspecified, this would put down "freaking annoying" as the default.

## #3 - 2011-07-15 11:04 - Jan Klopper

When would the default value kick in?

When the original argument cannot be resolved / is None?

Or when it's an empty string?

## #4 - 2011-07-15 11:04 - Jan Klopper

- Assignee changed from Jan Klopper to Elmer de Looff

## #5 - 2011-07-15 11:19 - Elmer de Looff

- Assignee changed from Elmer de Looff to Jan Klopper

Definitely when it cannot be resolved.

I'm not sure about boolean truthness, placing zeroes would be tricky if we rely on truthiness.

#### #6 - 2011-09-23 15:20 - Elmer de Looff

- *Category set to TemplateParser*

#### #7 - 2012-01-10 18:06 - Elmer de Looff

- *Status changed from Feedback to Resolved*

- *% Done changed from 0 to 70*

This can now be achieved using `{{ if }}` statements in the templateparser.

```
<p>This is a demonstration of the template default. Kittens are:</p>
<ul>
  <li>Cute</li>
  <li>{{ if [behaviour] }} [behaviour|lolcat] {{ else }} rather annoying {{ endif }}</li>
</ul>
```

This is more verbose than a quick default, but mostly the default will involve more than a single statement, and this solution quickly becomes preferable and readable.

#### #8 - 2012-02-17 16:10 - Elmer de Looff

- *Status changed from Resolved to Closed*

- *% Done changed from 70 to 100*

Seems to be good.